

SIMATIC SIMATIC Computing

Benutzerhandbuch

Dieses Handbuch ist Bestandteil des Dokumentationspaketes mit der Bestellnummer: **6ES7 673-6CC01-8AA0**

Vorwort, Inhaltsverzeichnis	
Erste Schritte mit SIMATIC Computing	1
Produktübersicht	2
Einrichten der Software SIMATIC Computing	3
Zugreifen auf Daten mit SIMATIC Computing	4
Zugreifen auf Prozeßdaten mit dem Data Control	5
User Controls	6
S7 Diagnostic Buffer Control (DBuffer)	7
Entwerfen von einfachen Prozeß-Formularen im WinAC SoftContainer	8
Erstellen von Symboldateien mit dem Symboldatei-Konfigurator	9
Speicherbereiche von S7-Automatisierungssystemen	A
Eigenschaften und Methoden	B
Events	C
Einrichten der SIMATIC Computing Konfiguration	D
Einsetzen von SIMATIC Computing mit DCOM	E
Richtlinien zum Programmieren mit SIMATIC Computing	F
Zeichenfolgen für Control Engines	G
Index	

Sicherheitstechnische Hinweise

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise sind durch ein Warndreieck hervorgehoben und je nach Gefährungsgrad folgendermaßen dargestellt:



Gefahr

bedeutet, daß Tod, schwere Körperverletzung oder erheblicher Sachschaden eintreten **werden**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.



Warnung

bedeutet, daß Tod, schwere Körperverletzung oder erheblicher Sachschaden eintreten **können**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.



Vorsicht

bedeutet, daß eine leichte Körperverletzung oder ein Sachschaden eintreten können, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Hinweis

ist eine wichtige Information über das Produkt, die Handhabung des Produktes oder den jeweiligen Teil der Dokumentation, auf den besonders aufmerksam gemacht werden soll.

Qualifiziertes Personal

Inbetriebsetzung und Betrieb eines Gerätes dürfen nur von **qualifiziertem Personal** vorgenommen werden. Qualifiziertes Personal im Sinne der sicherheitstechnischen Hinweise dieses Handbuchs sind Personen, die die Berechtigung haben, Geräte, Systeme und Stromkreise gemäß den Standards der Sicherheitstechnik in Betrieb zu nehmen, zu erden und zu kennzeichnen.

Bestimmungsgemäßer Gebrauch

Beachten Sie folgendes:



Warnung

Das Gerät darf nur für die im Katalog und in der technischen Beschreibung vorgesehenen Einsatzfälle und nur in Verbindung mit von Siemens empfohlenen bzw. zugelassenen Fremdgeräten und -komponenten verwendet werden.

Der einwandfreie und sichere Betrieb des Produktes setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung und Montage sowie sorgfältige Bedienung und Instandhaltung voraus.

Warenzeichen

SIMATIC®, SIMATIC HMI® und SIMATIC NET® sind eingetragene Warenzeichen der SIEMENS AG.

Die übrigen Bezeichnungen in dieser Schrift können Warenzeichen sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen können.

Copyright Siemens AG 1999–2000 All rights reserved

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts ist nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte vorbehalten, insbesondere für den Fall der Patenterteilung oder GM-Eintragung

Siemens AG
Bereich Automatisierungs- und Antriebstechnik
Geschäftsgebiet Industrie-Automatisierungssysteme
Postfach 4848, D- 90327 Nürnberg

Haftungsausschluß

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so daß wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, und notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten. Für Verbesserungsvorschläge sind wir dankbar.

© Siemens AG 1999–2000
Technische Änderungen bleiben vorbehalten.

Vorwort

Die Software SIMATIC Computing bedient sich der Microsoft ActiveX-Technologie (auch OLE genannt), um Ihnen Zugriff auf die Daten Ihrer Control Engine zu gewähren. Die Software SIMATIC Computing umfaßt folgende Bestandteile:

- Eine Reihe von SIMATIC Controls, bei denen es sich um ActiveX- oder OCX-Controls (OLE Custom Controls) zum Zugreifen auf die Control Engine handelt.
- Einen OPC-Server (OLE for Process Control), der anderen OPC-Anwendungen den Zugriff auf Daten in der Control Engine gewährt (z.B. WinLC von WinAC Basis oder CPU 416-2 DP ISA von WinAC Pro).
- Einen Symboldatei-Konfigurator zum Erstellen von Symboldateien, wodurch die symbolische Adressierung und der entfernte Zugriff auf mehrere Control Engines möglich wird.
- Ein Konfigurationswerkzeug zum Einrichten des entfernten Zugriffs.
- Einen OLE-Container (SoftContainer) zum Anlegen von Prozeß-Formularen mit SIMATIC Controls.

Hinweis

In der Software SIMATIC Computing bezieht sich der Begriff "Control Engine" auf Prozessoren oder Programme, die Daten, die einen Prozeß oder eine Maschine steuern, verwalten und bearbeiten. Bei der Control Engine kann es sich um Software oder um Hardware handeln.

WinAC Basis verfügt über den Windows Logic Controller (WinLC) als Control Engine. WinAC Pro hat die Slot-PLC als Control Engine. (Der Begriff "Slot-PLC" in diesem Handbuch bezieht sich auf eine Slot-PLC wie die CPU 416-2 DP ISA oder die CPU 416-2 DP ISA Lite. Im Handbuch fällt die CPU 416-2 DP ISA Lite unter die Beschreibung der CPU 416-2 DP ISA). Die ActiveX Controls in SIMATIC Computing kommunizieren mit diesen Control Engines sowie mit anderen SIMATIC S7-Automatisierungssystemen.

Leserkreis

Dieses Handbuch wendet sich an Ingenieure, Programmierer und Wartungspersonal mit allgemeinen Kenntnissen über Automatisierungssysteme (SPS).

Umfang des Handbuchs

Dieses Handbuch beschreibt die Funktionen und die Bedienung der Software SIMATIC Computing, Version 3.0.

Verwendung des Handbuchs

Dieses Handbuch enthält Informationen, die für unterschiedliche Zielgruppen gedacht sind. Es gibt nicht nur zwei Methoden zum Zugreifen auf Prozeßdaten (entweder über die ActiveX Controls oder über die OPC-Schnittstelle), sondern es gibt auch verschiedene Schwierigkeitsgrade für jede Methode. Sie können entweder die zur Verfügung gestellten Controls verwenden oder Sie können Programme schreiben, die diese Controls enthalten.

Wenn Sie mit ActiveX (OCX) Controls in einer Container-Anwendung wie Visual Basic arbeiten möchten, erhalten Sie ausführliche Informationen hierzu in den ersten Schritten (Kapitel 1) und in der Produktübersicht (Kapitel 2).

Die Kapitel zu den jeweiligen SIMATIC Controls enthalten Informationen zum Konfigurieren der Controls. Anhang B beschreibt die Eigenschaften und Methoden der Controls, und Anhang C beschreibt die Ereignisse.

Wenn Sie mit der OPC-Schnittstelle arbeiten möchten:

- Wenn Sie eine bestehende OPC-Client-Anwendung (von einem anderen Hersteller) mit den WinAC-Produkten verbinden möchten, erfahren Sie in der Produktübersicht (Kapitel 2) den Namen des OPC-Servers.
- Wenn Sie eine Client-Anwendung entwerfen möchten, die Sie zusammen mit den WinAC-Produkten einsetzen wollen, erfahren Sie in der OPC-Dokumentation (*OLE for Process Control Data Access Standard, Version 2.0*, herausgegeben von der OPC Foundation) nähere Informationen hierzu.

Weitere Handbücher

Informationen zu SIMATIC Computing finden Sie außerdem in der Online-Hilfe der Software. Zusätzliche Informationen bieten Ihnen die folgenden Handbücher:

Titel	Inhalt
Programmierhandbuch Programmmentwurf S7-300/400	Dieses Handbuch vermittelt das grundlegende Wissen über den Aufbau des Betriebssystems und über das Entwerfen eines Anwenderprogramms in WinLC. Arbeiten Sie mit diesem Handbuch, wenn Sie mit der Automatisierungssoftware STEP 7 ein Anwenderprogramm entwerfen.
OPC Server Interface Manual	Dieses Handbuch beschreibt das OPC Server Interface, das im Lieferumfang der Software SIMATIC Computing enthalten ist.
Windows Logic Controller (WinLC) Benutzerhandbuch	Dieses Handbuch vermittelt das grundlegende Wissen über die Funktionsweise und die Leistungsdaten des WinLC Controllers.
WinAC Controlling und CPU 416-2 DP ISA Hardware- und Installationshandbuch	Dieses Handbuch vermittelt das grundlegende Wissen über die Funktionsweise und die technischen Daten der CPU 416-2 DP ISA.

Zusätzliche Unterstützung

Falls Sie technische Fragen haben, die in diesem oder den anderen STEP 7-Handbüchern nicht behandelt werden, oder falls Sie Informationen zum Bestellen zusätzlicher Dokumentation oder Ausrüstung bzw. zu Schulungen benötigen, wenden Sie sich bitte an Ihre Siemens-Vertretung.

Siemens Kundendienst in Nordamerika:

- Telefon:
 - (609) 734-6500
 - (609) 734-3530
- E-Mail:
 - ISBU.Hotline@sea.siemens.com
 - simatic.hotline@sea.siemens.com
- Internet:
 - <http://www.aut.sea.siemens.com/winac/>
 - <http://www.aut.sea.siemens.com/simatic/support/index.htm>
 - http://www.ad.siemens.de/support/html_76/index.shtml
 - <http://www.sea.siemens.com/industrialsoftware/>

Siemens Kundendienst in Europa:

- Telefon: ++49 (0) 911 895 7000
- Fax: ++49 (0) 911 895 7001
- E-Mail: simatic.support@nbgm.siemens.de
- Internet: <http://www.ad.siemens.de/simatic-cs>

Inhaltsverzeichnis

1	Erste Schritte mit SIMATIC Computing	1-1
1.1	Übersicht	1-2
1.2	Erstellen eines Beispiels für ein Bedienpanel	1-4
1.3	Verbinden von Fremd-Controls mit einem Data Control	1-12
1.4	Einsetzen von SIMATIC Computing mit Microsoft Excel	1-15
1.5	Arbeiten mit dem SoftContainer von SIMATIC Computing	1-19
2	Produktübersicht	2-1
2.1	Produktübersicht	2-2
2.2	Zugreifen auf Prozeßdaten mittels ActiveX Control	2-4
2.3	Herstellen einer Verbindung zum Prozeß mit dem OPC-Server in WinAC	2-6
3	Einrichten der Software SIMATIC Computing	3-1
3.1	Übersicht	3-2
3.2	Autorisierung	3-3
3.3	Installieren und Deinstallieren der Software SIMATIC Computing	3-5
3.4	Verbinden von SIMATIC Computing mit einer Slot-PLC oder einer Kommunikationskarte	3-7
4	Zugreifen auf Daten mit SIMATIC Computing	4-1
4.1	Zugreifen auf Daten in Control Engines	4-2
4.2	Zugreifen auf eine lokale Control Engine	4-4
4.3	Zugreifen auf eine entfernte Control Engine	4-5
4.4	Kommunizieren mit mehreren Control Engines	4-6
5	Zugreifen auf Prozeßdaten mit dem Data Control	5-1
5.1	Verbinden der SIMATIC Controls mit der Control Engine	5-2
5.2	Einrichten der Verbindungseigenschaften für das Data Control	5-3
5.3	Einstellen der Control Engine für das Data Control	5-4
5.4	Verbinden der ActiveX Controls mit der Control Engine	5-9
5.5	Filtern der Eigenschaften für die ActiveX Controls	5-13
5.6	Einrichten von anwenderspezifischen Ereignissen	5-15
5.7	Erstellen einer Verbindungstabelle	5-16
5.8	Beispielprogramm zum Erstellen einer Verbindungstabelle und einer Ereignistabelle	5-17
5.9	Beispielprogramm zum Reagieren auf Ereignisse	5-19
5.10	Beispielprogramme zum Lesen und Schreiben von Daten	5-23
5.11	Beispielprogramm zum Lesen und Schreiben von booleschen Daten ...	5-28

5.12	Eigenschaften, Methoden und Ereignisse des Data Control	5-29
6	User Controls	6-1
6.1	Verbinden der User Controls mit Prozeßdaten	6-2
6.2	Arbeiten mit dem Eigenschaftsdialog des Button Control	6-4
6.3	Eigenschaften und Methoden des Button Control	6-9
6.4	Ereignisse des Button Control	6-10
6.5	Arbeiten mit dem Eigenschaftsdialog des Edit Control	6-11
6.6	Eigenschaften und Methoden des Edit Control	6-18
6.7	Ereignisse des Edit Control	6-19
6.8	Fehlercodes des Edit Control	6-20
6.9	Arbeiten mit dem Eigenschaftsdialog des Label Control	6-21
6.10	Eigenschaften und Methoden des Label Control	6-26
6.11	Ereignisse des Label Control	6-26
6.12	Arbeiten mit dem Eigenschaftsdialog des Slider Control	6-27
6.13	Eigenschaften und Methoden des Slider Control	6-34
6.14	Ereignisse des Slider Control	6-35
7	S7 Diagnostic Buffer Control (DBuffer)	7-1
7.1	Zugreifen auf den S7-Diagnosepuffer	7-2
7.2	Konfigurieren des DBuffer Control	7-4
7.3	Eigenschaften und Methoden des DBuffer Control	7-7
8	Entwerfen von einfachen Prozeß-Formularen im WinAC SoftContainer	8-1
8.1	Starten des SIMATIC Computing SoftContainer	8-2
8.2	Erstellen eines Prozeß-Formulars	8-4
8.3	Wechseln von der Betriebsart Design in die Betriebsart RUN	8-6
8.4	Speichern Ihres Prozeß-Formulars	8-8
9	Erstellen von Symboldateien mit dem Symboldatei-Konfigurator	9-1
9.1	Verbinden mehrerer Control Engines über DCOM	9-2
9.2	Zugreifen auf Daten in der Control Engine über Symbole	9-5
9.3	Erstellen einer Symboldatei	9-6
9.4	Einrichten einer Symboldatei für lokalen oder entfernten Zugriff auf eine Control Engine	9-10
9.5	Ändern des symbolischen Namens der Control Engine im Symboldatei-Editor	9-13

A	Speicherbereiche von S7-Automatisierungssystemen	A-1
A.1	Speicherbereiche von S7-Automatisierungssystemen	A-2
A.2	Zugreifen auf die S7-Datentypen	A-3
A.3	Beschreibungen der S7-Datentypen	A-6
B	Eigenschaften und Methoden	B-1
B.1	Methode AboutBox	B-1
B.2	Eigenschaft Activated	B-1
B.3	Eigenschaft Alignment	B-2
B.4	Eigenschaft Appearance	B-3
B.5	Eigenschaft AutoConnect	B-3
B.6	Eigenschaft AutoConnectTimeout	B-4
B.7	Eigenschaft BackColor	B-5
B.8	Eigenschaft bDiagBuffOK	B-5
B.9	Eigenschaft bEngineConnected	B-6
B.10	Eigenschaft BorderStyle	B-6
B.11	Eigenschaft Caption	B-7
B.12	Methode Connect	B-7
B.13	Methode ConnectName	B-8
B.14	Methode ConnectObject	B-9
B.15	Eigenschaft ControlEngine	B-10
B.16	Eigenschaft DataFormat	B-11
B.17	Eigenschaft DefaultDeadband	B-12
B.18	Eigenschaft DefaultUpdateRate	B-13
B.19	Eigenschaft Direction	B-14
B.20	Methode Disconnect	B-14
B.21	Eigenschaft DisplayFormatButtons	B-15
B.22	Eigenschaft DisplayHelpButton	B-15
B.23	Eigenschaft DisplayHelpOnEventButton	B-16
B.24	Eigenschaft DisplayLowerPanel	B-17
B.25	Eigenschaft DisplayUpdateButton	B-17
B.26	Eigenschaft DisplayUpperPanel	B-18
B.27	Eigenschaft DisplayValue	B-18
B.28	Eigenschaft Enabled	B-19
B.29	Eigenschaft EnableSort	B-19
B.30	Eigenschaft Factor	B-20
B.31	Eigenschaft FalseCaption	B-21

B.32	Eigenschaft FalseColor	B-21
B.33	Eigenschaft FalsePicture	B-22
B.34	Eigenschaft Font	B-22
B.35	Eigenschaft ForeColor	B-23
B.36	Eigenschaft FormatDisplay	B-23
B.37	Eigenschaft KnobHeight	B-24
B.38	Eigenschaft KnobPicture	B-24
B.39	Eigenschaft KnobWidth	B-25
B.40	Eigenschaft LargeChange	B-25
B.41	Eigenschaft Locked	B-26
B.42	Eigenschaften Max und Min	B-26
B.43	Eigenschaft MultipleEngines	B-27
B.44	Eigenschaft Offset	B-27
B.45	Eigenschaft PCName	B-28
B.46	Eigenschaft Picture	B-28
B.47	Methode PopUpHelp	B-29
B.48	Methode PopUpHelpOnEvent	B-29
B.49	Eigenschaft Precision	B-30
B.50	Methode PropertyChangedName	B-30
B.51	Methode PropertyChangedObject	B-31
B.52	Eigenschaft PushButton	B-32
B.53	Eigenschaften RawMax und RawMin	B-32
B.54	Methode ReadMultiVariables	B-33
B.55	Methode ReadVariable	B-33
B.56	Eigenschaft ScaleMode	B-34
B.57	Methode SelectEvent	B-35
B.58	Eigenschaft ShowErrorBoxes	B-36
B.59	Eigenschaft ShowMinMax	B-36
B.60	Eigenschaft SmallChange	B-37
B.61	Eigenschaft StretchMode	B-38
B.62	Eigenschaft Style	B-39
B.63	Eigenschaft TagSource	B-39
B.64	Eigenschaft Text	B-40
B.65	Eigenschaft Ticks	B-40
B.66	Eigenschaft TrueCaption	B-40
B.67	Eigenschaft TrueColor	B-41

B.68	Eigenschaft TruePicture	B-42
B.69	Methode Update	B-42
B.70	Eigenschaft Value	B-43
B.71	Eigenschaft WriteMode	B-43
B.72	Methode WriteNow	B-44
B.73	Methode WriteMultiVariables	B-44
B.74	Methode WriteVariable	B-45
B.75	Eigenschaft ZeroPad	B-46
C	Events	C-1
C.1	Ereignis Change	C-1
C.2	Ereignis Click	C-1
C.3	Ereignis ConnectionError	C-1
C.4	Ereignis DblClick	C-2
C.5	Ereignis Error	C-2
C.6	Ereignis KeyDown	C-3
C.7	Ereignis KeyPress	C-5
C.8	Ereignis KeyUp	C-5
C.9	Ereignis MouseDown	C-6
C.10	Ereignis MouseMove	C-7
C.11	Ereignis MouseUp	C-8
C.12	Ereignis ValueChanged	C-9
D	Einrichten der SIMATIC Computing Konfiguration	D-1
D.1	Konfigurieren der OPC-Verbindung	D-2
D.2	Auswählen der Sprache	D-5
D.3	Auswählen der Control Engine für ältere Programme	D-6
D.4	Einrichten der Kommunikation in der Anwendung "PG/PC-Schnittstelle einstellen"	D-7
E	Einsetzen von SIMATIC Computing mit DCOM	E-1
E.1	Entfernter Zugriff über DCOM	E-2
E.2	Einrichten der Berechtigungen für den Server-Computer	E-4
E.3	Einrichten der Berechtigungen für den Client-Computer	E-14
E.4	Fehlerbehebung	E-20

F	Richtlinien zum Programmieren mit SIMATIC Computing	F-1
F.1	Richtlinien für Fremd-Container	F-2
F.2	Richtlinien zum Programmieren	F-4
F.3	Richtlinien zum Erstellen von anwenderspezifischen ActiveX Controls ..	F-6
F.4	Einsetzen eines anwenderspezifischen ActiveX Control mit einem Data Control	F-7
F.5	Bekannte Probleme in SIMATIC Computing Version 3	F-10
G	Zeichenfolgen für Control Engines	G-1
	Index	Index-1

Erste Schritte mit SIMATIC Computing

1

Kapitelübersicht

Die Software SIMATIC Computing bietet Ihnen verschiedene Möglichkeiten, um mit Daten aus einer Control Engine wie einer S7-CPU, dem Windows Logic Controller (WinLC) von WinAC Basis oder einer Slot-PLC wie der CPU 416-2 DP ISA von WinAC Pro, zu arbeiten.

Dieses Kapitel soll Sie anhand von einfachen Programmierbeispielen mit der Leistungsfähigkeit und Flexibilität der ActiveX Controls in SIMATIC Computing vertraut machen. Die Beispielprogramme finden Sie in folgendem Verzeichnis in dem Laufwerk, in dem Sie die Software SIMATIC Computing installiert haben:

[C:] \Siemens \WinAC \Examples



Warnung

Nachdem Sie der Eigenschaft Value eines SIMATIC oder Fremd-Control eine Variable zugeordnet haben, kann das Control auf Prozeßdaten zugreifen. Wenn Sie den Wert ändern, der von einem Control angezeigt wird, ändern Sie den Wert im tatsächlichen Prozeß. Verbinden Sie dieses Beispiel nicht mit einer Control Engine, die an Maschinen angeschlossen ist.

Durch das Ändern von Prozeßdaten können Sie unvorhersehbare Reaktionen im Prozeß auslösen, die zu Tod, schwerer Körperverletzung und/oder Sachschaden führen können.

Gehen Sie vorsichtig vor und achten Sie darauf, daß Sie nicht auf Daten zugreifen, die unerwartete Reaktionen in den gesteuerten Geräte hervorrufen könnten. Installieren Sie einen physikalischen NOT-AUS-Schaltkreis für die Maschine bzw. den Prozeß.

Kapitel	Beschreibung	Seite
1.1	Übersicht	1-2
1.2	Erstellen eines Beispiels für ein Bedienpanel	1-4
1.3	Verbinden von Fremd-Controls mit einem Data Control	1-12
1.4	Einsetzen von SIMATIC Computing mit Microsoft Excel	1-15
1.5	Arbeiten mit dem SoftContainer von SIMATIC Computing	1-19

1.1 Übersicht

In SIMATIC Computing können Sie nicht nur einfach auf die Daten in der Control Engine zugreifen, sondern Sie haben verschiedene Möglichkeiten, wie Sie auf die Daten zugreifen und wie Sie die Daten bearbeiten.

Die Beispiele in diesem Kapitel zeigen verschiedene Einsatzarten für die ActiveX Controls in SIMATIC Computing. Wie Sie in Bild 1-1 sehen, bietet dieses Kapitel Beispiele für Unterprogramme für folgende Anwendungen:

- Erstellen einer Bedienoberfläche: Sie können die SIMATIC Controls in Fremd-Containern (z.B. Microsoft Visual Basic) einsetzen, um ein Bedienpanel zu erstellen (siehe Abschnitt 1.2). (Mit diesem Panel können Sie die anderen Beispielprogramme in diesem Kapitel testen.)
- Einsetzen eines Standard-ActiveX-Control: Sie können auch mit einem Standard-Control (z.B. ein Label Control aus Visual Basic) auf Daten in der Control Engine zugreifen (siehe Abschnitt 1.3).
- Laden von Daten aus der Control Engine in Standard-Softwarepakete: Sie können Daten in eine Anwendung von Microsoft Office laden (z.B. Microsoft Excel) (siehe Abschnitt 1.4).

Anstatt mit einem Fremd-Container zu arbeiten (Abschnitt 1.2), können Sie den im Lieferumfang von SIMATIC Computing enthaltenen SoftContainer einsetzen, um ein einfaches Bedienpanel zu erstellen (siehe Abschnitt 1.5).

Die Beispielprogramme finden Sie in folgendem Verzeichnis in dem Laufwerk, in dem Sie die Software SIMATIC Computing installiert haben:

[C:] \Siemens \WinAC \Examples

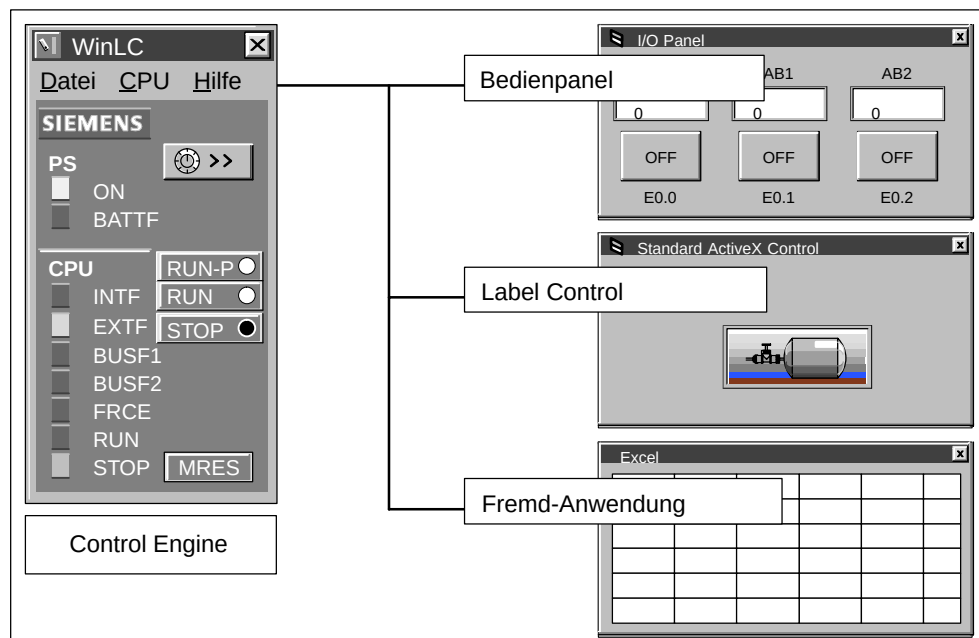


Bild 1-1 Zugreifen auf Daten in der Control Engine mit SIMATIC Computing

Beispielprogramm für die Anwendungsbeispiele

Bild 1-2 zeigt das Beispielprogramm, das von den Anwendungsbeispielen eingesetzt wird. Das Programm besteht aus folgender Logik:

- Ist das Eingangsbit 0.0 (E0.0) eingeschaltet, inkrementiert das Programm einen Wert, der in MB1 abgelegt ist, und überträgt den neuen Wert in AB0.
- Ist das Eingangsbit 0.1 (E0.1) eingeschaltet, dekrementiert das Programm einen Wert, der in MB3 abgelegt ist, und überträgt den neuen Wert in AB1.
- Ist das Eingangsbit 0.2 (E0.2) eingeschaltet, inkrementiert das Programm einen Wert, der in MB5 abgelegt ist, und überträgt den neuen Wert in AB2.

Legen Sie ein STEP 7 Projekt an und laden Sie das Programm in die Control Engine.

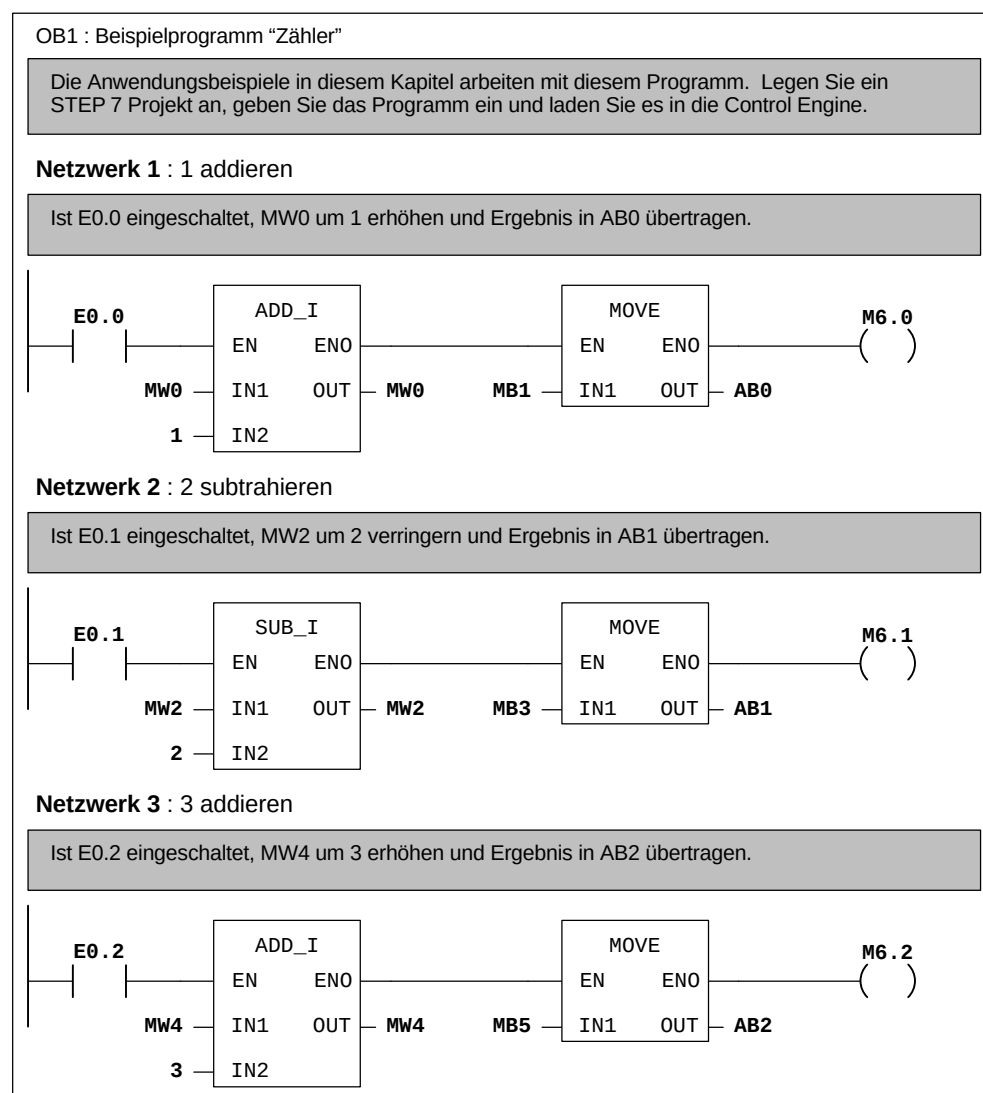


Bild 1-2 Beispielprogramm ("Zähler") für die Anwendungsbeispiele

1.2 Erstellen eines Beispiels für ein Bedienpanel

Das Data Control gewährt jedem ActiveX Container (z.B. Visual Basic 5.0) Zugriff auf die Daten in der Control Engine. Sie können mit den SIMATIC Controls von SIMATIC Computing in Visual Basic ein einfaches Bedienpanel erstellen, das mit dem Programm, das in der Control Engine läuft, kommuniziert.

Zum Erstellen dieser Beispielanwendung benötigen Sie folgendes:

- Microsoft Visual Basic 5 oder höher
- SIMATIC Controls aus SIMATIC Computing
- Control Engine: z.B. WinLC oder eine Slot-PLC wie die CPU 416-2 DP ISA
- Beispielprogramm (siehe Abschnitt 1.1)
- STEP 7 (zum Laden des Programms in die Control Engine)



Vorsicht

Wenn Sie die Zeitfunktion oder Haltepunkte in Visual Basic mit SIMATIC Computing verwenden, kann dies zu Fehlern führen, die evt. Ihren Computer bzw. die Anwendung zum Absturz bringen bzw. verriegeln können. Je nach Konfiguration kann die Anwendung dadurch die Kommunikation mit der Control Engine beenden. Dies kann unvorhersehbare Reaktionen im Prozeß auslösen, die zu Tod, schwerer Körperverletzung und/oder Sachschaden führen können.

Installieren Sie einen physikalischen NOT-AUS-Schaltkreis für die Maschine bzw. den Prozeß.

Timer in VB: Bei der Timer-Funktion in Visual Basic Version 5 kann ein Timer den in der gleichen Einweg-Bearbeitung befindlichen Code unterbrechen, wodurch Fehler mit schwerwiegenden Folgen auftreten können. Beachten Sie folgende Richtlinien, wenn Sie VB-Timer mit SIMATIC Computing einsetzen:

- Deaktivieren Sie die Timer immer im Unterprogramm Form_Unload. Andernfalls kann ein Timer ein Ereignis auslösen, während das VB-Programm geschlossen wird. Diese Bedingung kann einen Absturz bzw. eine Verriegelung Ihres Computers oder der Anwendung herbeiführen. Es kann auch sein, daß die Anwendung unsichtbar weiterläuft.
 - Wenn Sie den Timer im Unterprogramm Form_Load starten, kann es sein, daß das Zeitereignis auftritt, bevor die anderen Objekte instanziiert wurden. Um sicherzustellen, daß die Objekte korrekt instanziiert werden, starten Sie einen Timer im Unterprogramm Form_Load immer mit einem großen Intervall (z.B. 1 oder 2 Sekunden). Nachfolgende Timer-Intervalle können kürzer sein.
-

Einfügen der SIMATIC Controls in die Toolbox für Visual Basic

Gehen Sie folgendermaßen vor, um das Bedienpanel zu erstellen:

1. Öffnen Sie ein Projekt in Visual Basic:
 - Wählen Sie den Menübefehl **Datei > Neues Projekt**. Daraufhin wird das Dialogfeld "Neues Projekt" angezeigt.
 - Wählen Sie das Symbol "Standard EXE" und klicken Sie auf die Schaltfläche "Öffnen".
2. Wählen Sie den Menübefehl **Projekt > Komponenten**. Daraufhin wird das Dialogfeld "Komponenten" angezeigt.
3. Im Dialogfeld "Komponenten" wählen Sie die folgenden SIMATIC Controls (siehe Bild 1-3):
 - Data Control (Siemens SIMATIC DataControl)
 - Panel Control (Siemens S7 Panel Control, wird mit WinLC oder einer Slot-PLC geliefert)
 - Diagnosepuffer (Siemens WinAC Diagnostic Buffer Control)
 - User Controls (Siemens_WinAC_UserControls) Die Symbole für das Button, Label, Slider und Edit Control sind im Register "Icon" enthalten.
4. Wählen Sie die Schaltfläche "Übernehmen". Die markierten SIMATIC Controls erscheinen in der Toolbox in Visual Basic. Klicken Sie auf "OK", um das Dialogfeld "Komponenten" zu schließen.

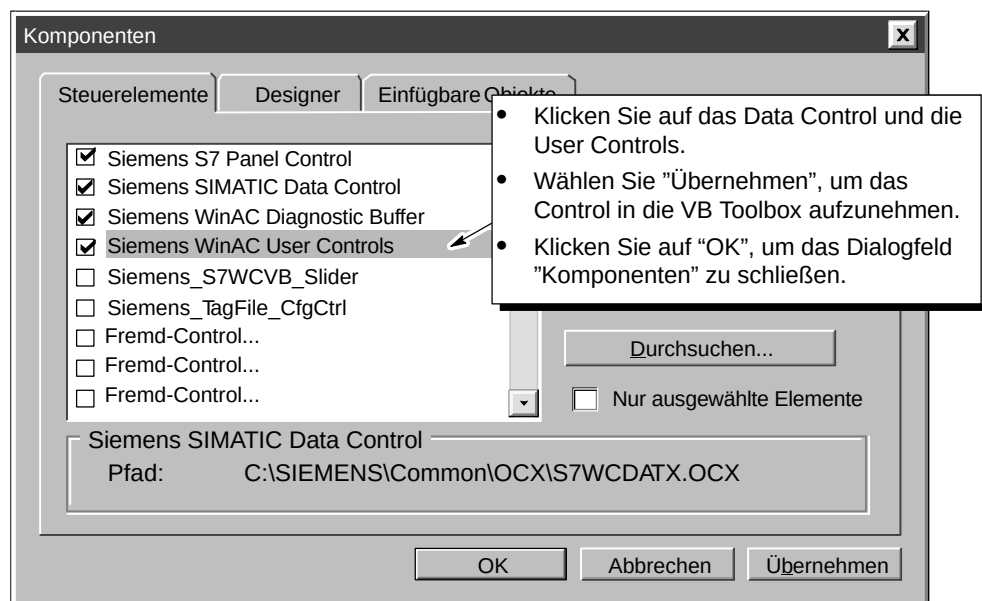


Bild 1-3 Aufnehmen von SIMATIC Controls in die VB Toolbox

Erstellen eines Arbeitsblatts in VB für das Bedienpanel

1. Fügen Sie ein Data Control, drei Edit Controls und drei Button Controls in das Arbeitsblatt in Visual Basic ein (siehe Bild 1-4).
2. Erstellen Sie VB Label Controls, um die Adresse, die Sie den einzelnen Controls zugewiesen haben, anzuzeigen (siehe Bild 1-4).

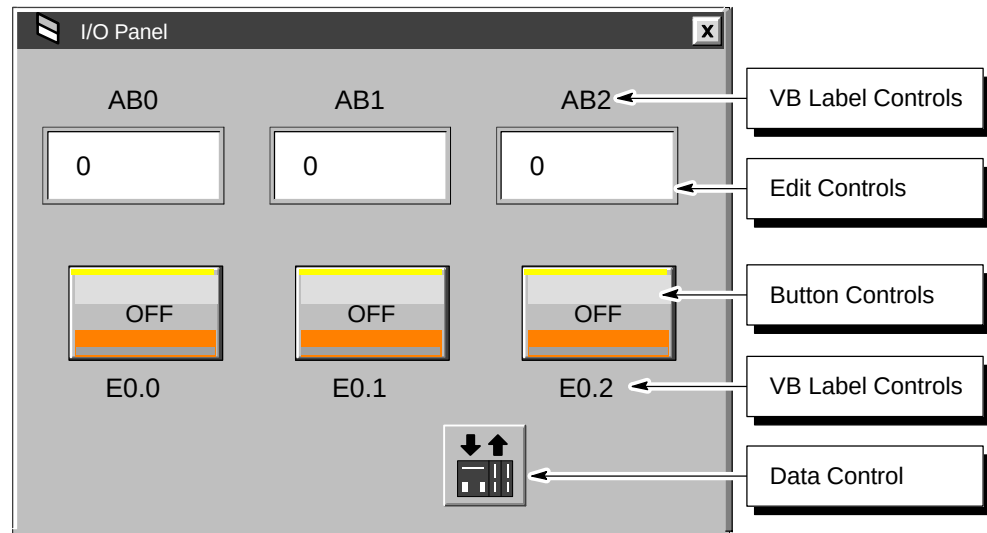


Bild 1-4 Beispiel für ein in Visual Basic erstelltes Bedienpanel

Zuordnen von Variablen in der Control Engine zu den SIMATIC Controls

Damit Sie die SIMATIC oder Fremd-Controls mit den Prozeßdaten in der Control Engine verbinden können, müssen Sie der Eigenschaft Value (oder anderen Eigenschaften) der Controls eine Variable (Adresse im Speicher der Control Engine) zuweisen. Die Variablen in der Control Engine ordnen Sie im Dialogfeld "Eigenschaften" im Register "Connection" zu. Sie können der Eigenschaft Value eines Control die Variable nicht über die Eigenschaftsliste des Control selbst zuordnen.

Gehen Sie folgendermaßen vor, um den SIMATIC Controls Variablen zuzuordnen:

1. Markieren Sie das Data Control und klicken Sie mit der rechten Maustaste, um das Popup-Menü aufzurufen. Im Popup-Menü wählen Sie den Befehl **Eigenschaften**, um das Dialogfeld "Eigenschaften" des Data Control anzuzeigen.
2. Öffnen Sie das Register "Connections". Klicken Sie auf das Symbol "+", um die Liste der Controls aufzuklappen.
3. Wählen Sie das Control und klicken Sie auf das Symbol "+", um die Liste der Eigenschaften aufzuklappen (siehe Bild 1-5).

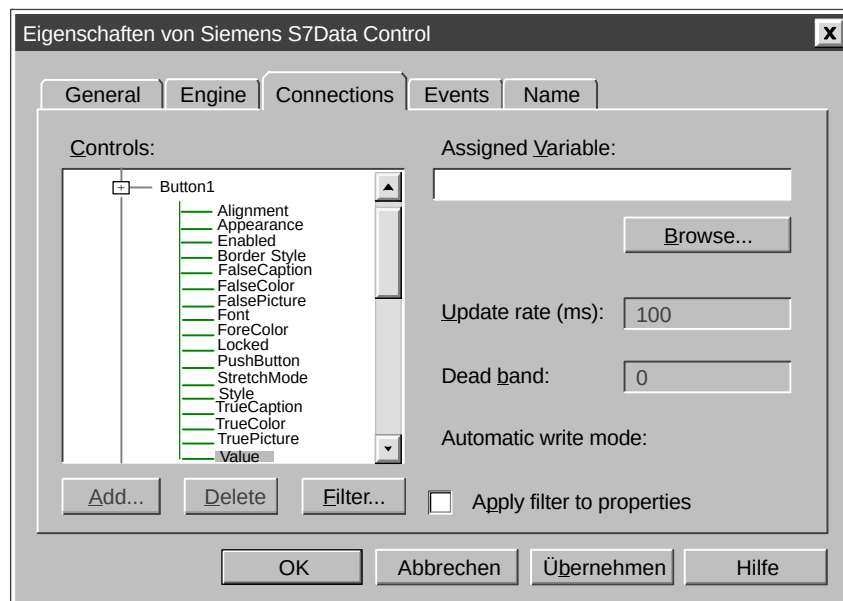


Bild 1-5 Anzeigen der Liste der Eigenschaften

4. Wählen Sie die Schaltfläche "Filter" (siehe Bild1-6).

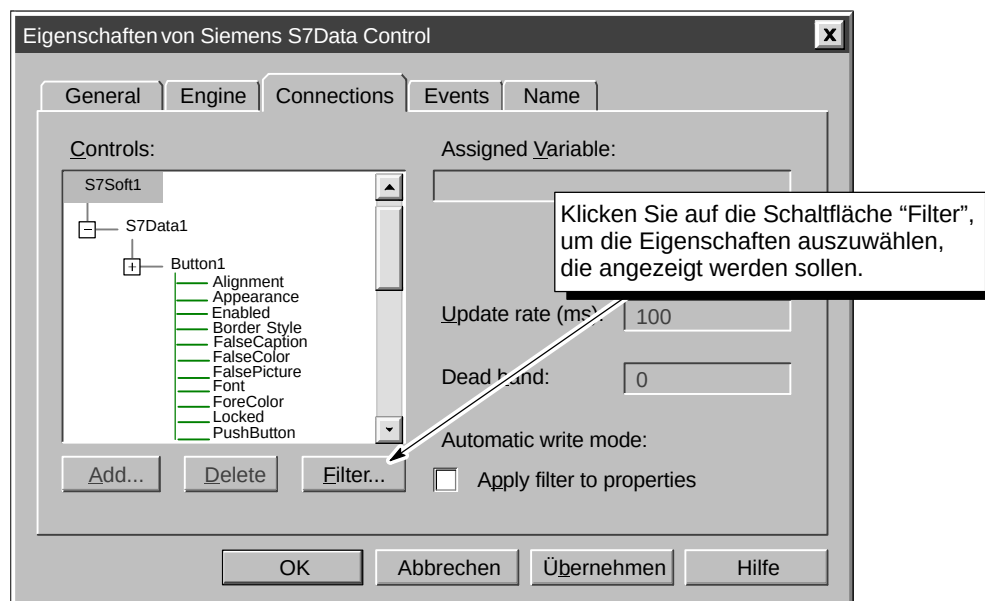


Bild 1-6 Schaltfläche "Filter"

5. Geben Sie die Eigenschaften, die angezeigt werden sollen, ein und wählen Sie die Schaltfläche "Add" (siehe Bild1-7). Mit der Schaltfläche "Edit" können Sie Einträge korrigieren und mit der Schaltfläche "Delete" löschen Sie Einträge.

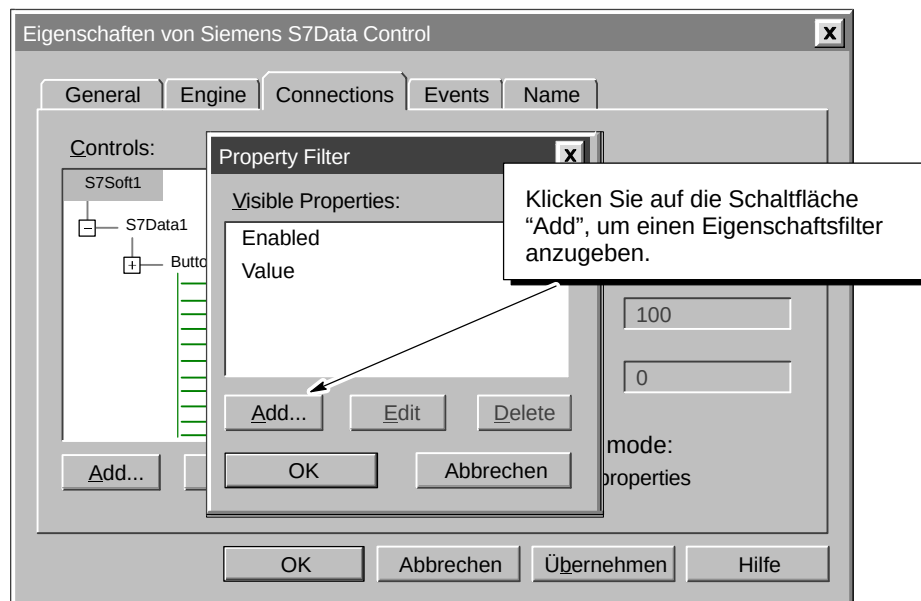


Bild 1-7 Hinzufügen von Eigenschaften zum Filter

6. Aktivieren Sie das Kontrollkästchen "Apply filter to properties" (siehe Bild1-8), um die gefilterten Eigenschaften anzuzeigen. Mit dem Kontrollkästchen "Apply filter to properties" können Sie den Filter ein- und ausschalten.

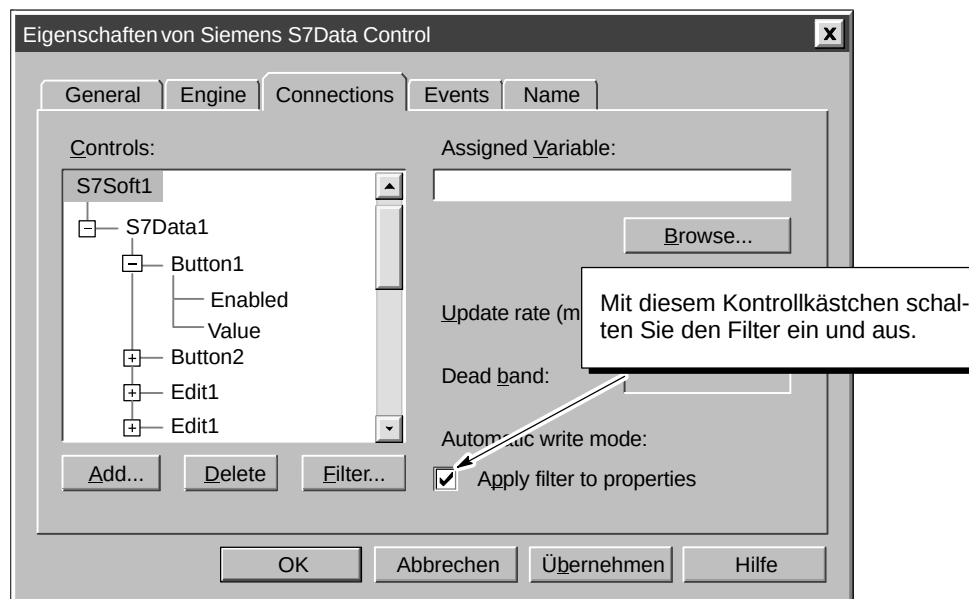


Bild 1-8 Anwenden des Filters auf die Liste der Eigenschaften

7. Wählen Sie die Eigenschaft Value des Control (siehe Bild1-9).

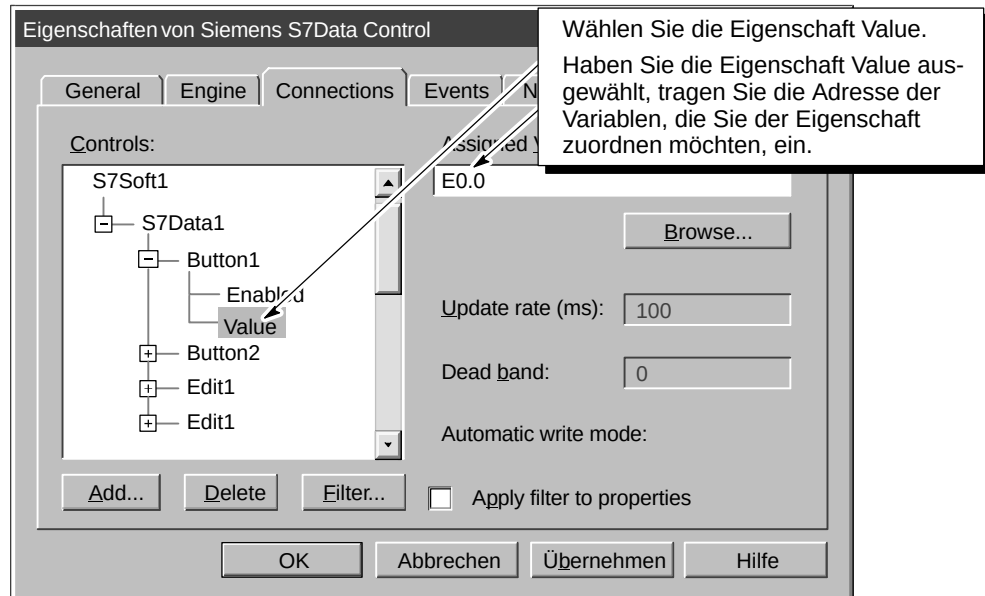


Bild 1-9 Zuordnen einer Variablen in der Control Engine zu einer Eigenschaft eines Control

8. Ordnen Sie die Variablen (Adressen im Speicher der Control Engine) zu den SIMATIC Controls wie in Tabelle 1-1 beschrieben zu.
9. Wählen Sie die Schaltfläche "Übernehmen", um die zugewiesenen Variablen einzugeben.

Tabelle 1-1 Zuordnen von Beispielandressen zu den SIMATIC Controls

Control	Adresse	Beschreibung
Edit1	AB0	Ausgabewert des ersten Zählers
Edit2	AB1	Ausgabewert des zweiten Zählers
Edit3	AB2	Ausgabewert des dritten Zählers
Button1	E0.0	Freigabebit für den ersten Zähler
Button2	E0.1	Freigabebit für den zweiten Zähler
Button3	E0.2	Freigabebit für den dritten Zähler

Einstellen einer Control Engine

Gehen Sie folgendermaßen vor, um das Data Control für eine Verbindung mit einer bestimmten Control Engine einzurichten:

1. Öffnen Sie das Register "Engine", um die Control Engine einzurichten (siehe Bild 1-10).
2. Wählen Sie die Option "Direct Connect" und geben Sie die Control Engine ein, z.B. **WinLC** oder **wcS7=3** (für eine Slot-PLC wie die CPU 416-2 DP ISA). Wählen Sie die Schaltfläche "Übernehmen", um die Daten zu übernehmen und bestätigen Sie mit "OK", um das Dialogfeld zu schließen.

Hinweis

wcS7=3 ist identisch mit S7DosIntf/MPI=3, das noch aus Kompatibilitätsgründen aktiviert ist.

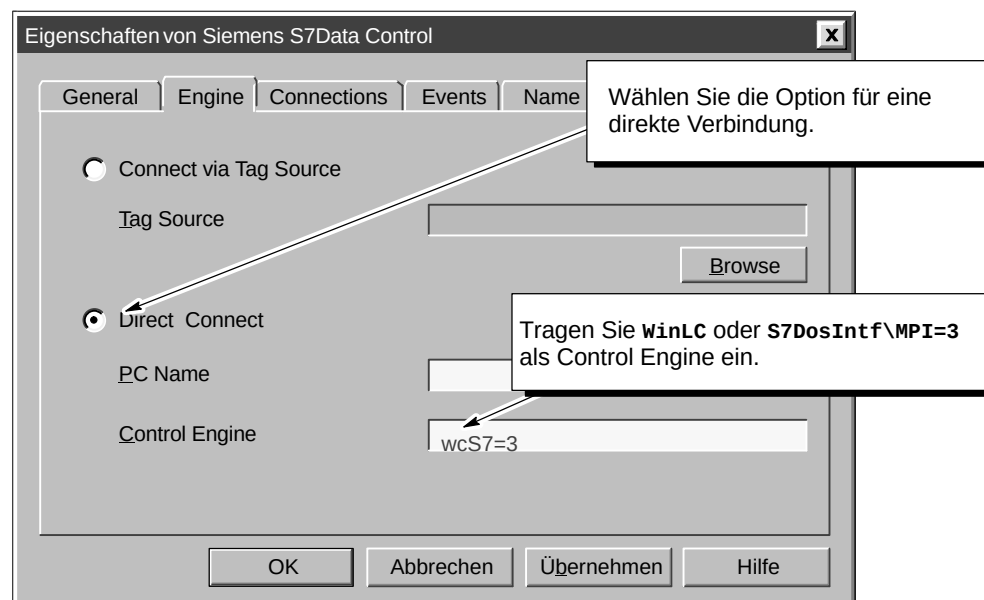


Bild 1-10 Verbinden des Data Control mit einer Control Engine

Ausführen des Programms für das Bedienpanel

Bevor Sie das Programm für das Bedienpanel ausführen, vergewissern Sie sich, daß die Control Engine das Beispielprogramm "Zähler" ausführt.

Hinweis

Ist die Control Engine (z.B. WinLC oder eine Slot-PLC wie die CPU 416-2 DP ISA) nicht aktiv, kann das Data Control keine Verbindung aufbauen. Bevor Sie Visual Basic in den Betriebszustand RUN versetzen, vergewissern Sie sich, daß die Control Engine läuft.

1. Wählen Sie den Menübefehl **Datei > Projekt speichern**, um das Programm zu speichern, bevor Sie in Visual Basic von der Betriebsart Design in die Betriebsart RUN wechseln.
2. Klicken Sie auf die Schaltfläche zum Starten oder wählen Sie den Menübefehl **Ausführen > Starten**, um Visual Basic von der Betriebsart Design in die Betriebsart RUN zu versetzen und das Programm für das Bedienpanel auszuführen.
3. Klicken Sie auf das Button Control für E0.0, um den ersten Zähler zu starten (siehe Bild 1-11).
 - Das Button Control ändert die Farbe, um den Zustand von E0.0 anzuzeigen.
 - Das Edit Control für AB0 zeigt den Zählwert an.
4. Klicken Sie auf das Button Control für E0.1, um den zweiten Zähler zu starten (siehe Bild 1-11).
 - Das Button Control ändert die Farbe, um den Zustand von PE0.1 anzuzeigen.
 - Das Edit Control für AB1 zeigt den Zählwert an.
5. Klicken Sie auf das Button Control für E0.2, um den dritten Zähler zu starten (siehe Bild 1-11).
 - Das Button Control ändert die Farbe, um den Zustand von E0.2 anzuzeigen.
 - Das Edit Control für AB2 zeigt den Zählwert an.

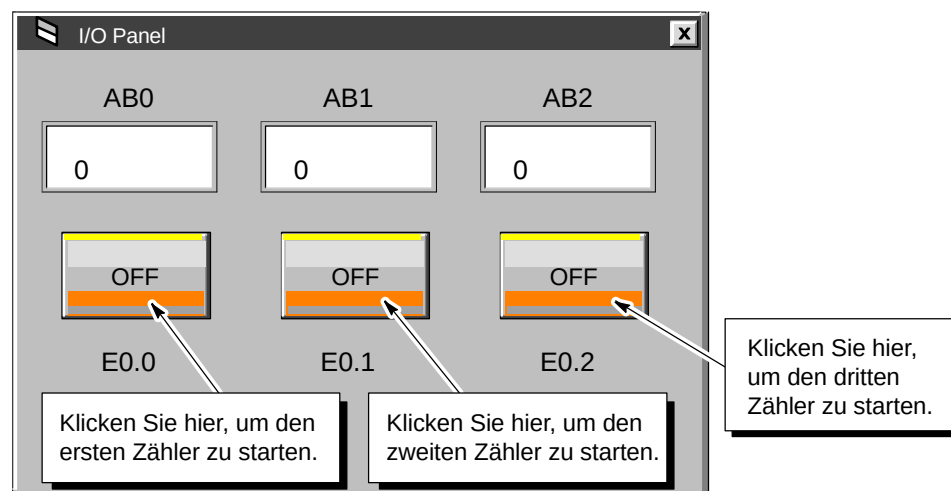


Bild 1-11 Bedienpanel

1.3 Verbinden von Fremd-Controls mit einem Data Control

Mit dem Data Control können Sie beliebige ActiveX Controls (z.B. das VB Label Control) mit den Daten in der Control Engine verbinden. Zum Erstellen dieser Beispielanwendung benötigen Sie folgendes:

- Microsoft Visual Basic 5 oder höher
 - Data Control aus SIMATIC Computing
 - Control Engine: z.B. WinLC oder eine Slot-PLC wie die CPU 416-2 DP ISA
 - Beispielprogramm (siehe Abschnitt 1.1)
 - STEP 7 (zum Laden des Programms in die Control Engine und zum Einschalten der Eingangsbits im Peripheriebereich des Beispielprogramms)
- Sie können die Eingangsbits im Peripheriebereich des Beispielprogramms, das in der Control Engine ausgeführt wird, auch mit der Beispielanwendung des Bedienpanel einschalten. Ausführliche Informationen zur Anwendung des Bedienpanel entnehmen Sie dem Abschnitt 1.2.

Erstellen eines VB Label Control zum Anzeigen eines Werts der Control Engine

Gehen Sie folgendermaßen vor, um das Data Control mit einem Label Control zu verbinden:

1. Öffnen Sie ein Projekt in Visual Basic: Wählen Sie den Menübefehl **Datei > Neues Projekt**, um das Dialogfeld "Neues Projekt" aufzurufen, wählen Sie dann das Symbol "Standard EXE" und klicken Sie auf die Schaltfläche zum Öffnen.
2. Nehmen Sie das Data Control in die VB Toolbox auf. Ausführliche Informationen zum Aufnehmen von Controls in die VB Toolbox entnehmen Sie dem Abschnitt 1.1 und dem Bild 1-12.

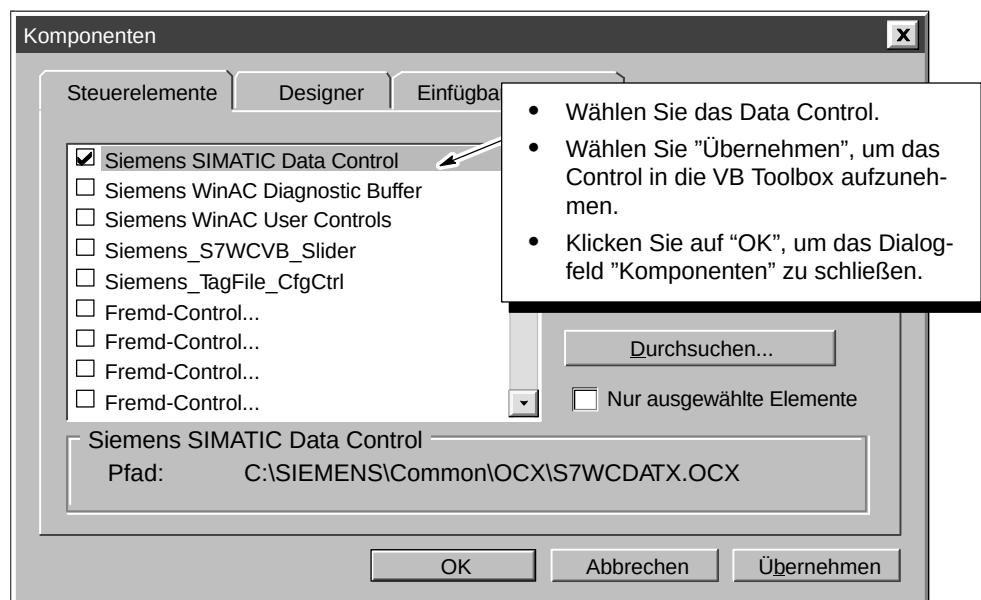


Bild 1-12 Aufnehmen des Data Control in die Visual Basic Toolbox

3. Fügen Sie ein Data Control in das Arbeitsblatt in Visual Basic ein (Informationen zum Einfügen von Controls in ein Arbeitsblatt in Visual Basic entnehmen Sie dem Abschnitt 1.1).
4. Fügen Sie ein VB Label Control in Ihr Arbeitsblatt ein. Stellen Sie die Eigenschaft Border Style auf "1-Fixed Single."
5. Markieren Sie das Data Control und klicken Sie mit der rechten Maustaste, um das Popup-Menü aufzurufen. Im Popup-Menü wählen Sie den Befehl **Eigenschaften**, um das Dialogfeld "Eigenschaften" des Data Control anzuzeigen.
6. Öffnen Sie im Dialogfeld "Eigenschaften" das Register "Connections". Klicken Sie auf das Symbol "+", um die Liste der Controls aufzuklappen.
7. Markieren Sie das Label1 Control und klicken Sie dessen Symbol "+", um die Liste der Eigenschaften aufzuklappen.
8. Wählen Sie die Eigenschaft Caption und tragen Sie **AB0** in das Feld "Assigned Variable" ein (siehe Bild 1-13). Klicken Sie nacheinander auf die Schaltflächen "Übernehmen" und "OK", um die Daten einzugeben und das Dialogfeld "Eigenschaften" zu schließen.

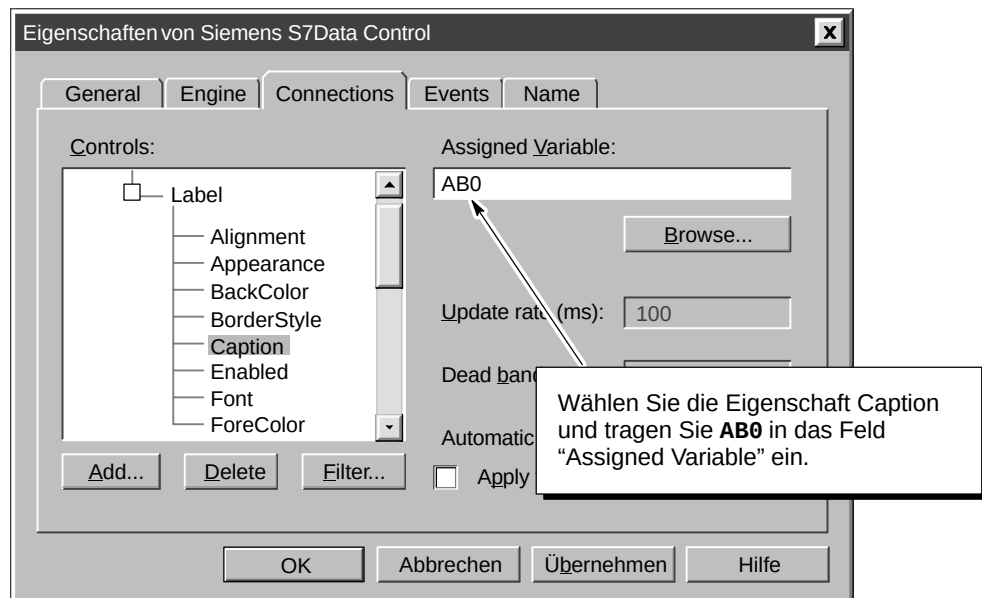


Bild 1-13 Zuordnen einer Variablen zur Eigenschaft Caption eines VB Label Control

Ausführen des Beispielprogramms für das Label Control

Speichern Sie das Programm, bevor Sie Visual Basic von der Betriebsart Design in RUN versetzen. Wird das Beispielprogramm ausgeführt, zeigt der Text des Label Control den Wert von AB0 in der Control Engine an.

Hinweis

Ist die Control Engine (z.B. WinLC oder eine Slot-PLC wie die CPU 416-2 DP ISA) nicht aktiv, kann das Data Control keine Verbindung aufbauen. Bevor Sie Visual Basic in den Betriebszustand RUN versetzen, vergewissern Sie sich, daß die Control Engine läuft.

Gehen Sie folgendermaßen vor, wenn Sie das Data Control für die Kommunikation mit der Control Engine und für die Ausführung des Beispielprogramms konfigurieren möchten.

1. Öffnen Sie das Register "Engine", um die Control Engine einzurichten (siehe Bild 1-14).
2. Wählen Sie die Option "Direct Connect" und geben Sie entweder **WinLC** oder **wcS7=3** (für eine Slot-PLC wie die CPU 416-2 DP ISA) für die Control Engine ein. Wählen Sie die Schaltfläche "Übernehmen", um die Daten zu übernehmen und bestätigen Sie mit "OK", um das Dialogfeld zu schließen.
3. Versetzen Sie Visual Basic von der Betriebsart Design in RUN, um das Beispielprogramm auszuführen.

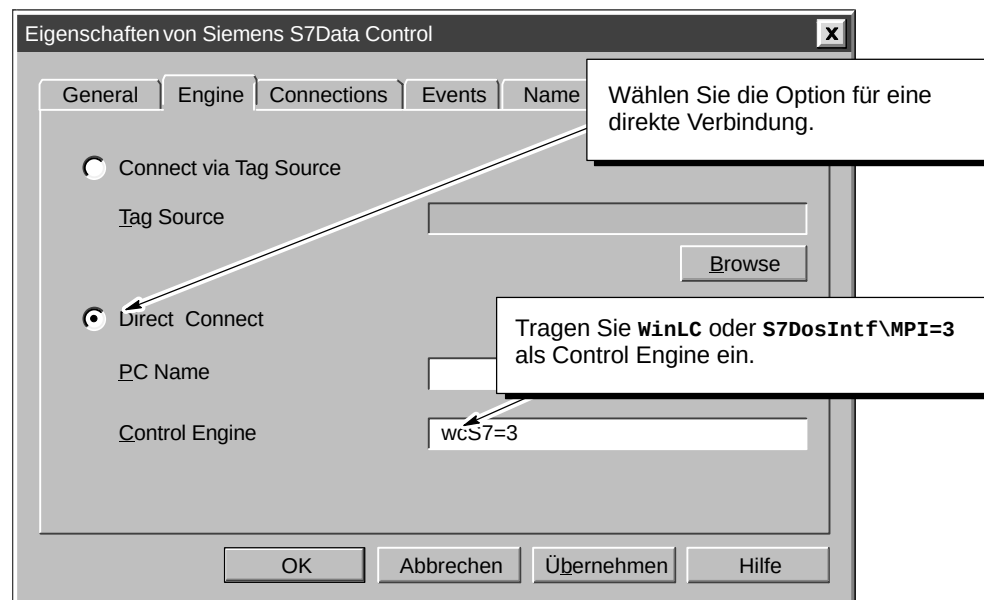


Bild 1-14 Verbinden mit der Control Engine (Beispiel: Label Control)

1.4 Einsetzen von SIMATIC Computing mit Microsoft Excel

Wenn Sie das Data Control in einer Excel-Tabelle einsetzen, können Sie auf die Werte in der Control Engine zugreifen. Zum Erstellen dieser Beispielanwendung benötigen Sie folgendes:

- Microsoft Excel 97 oder Excel 2000
- Control Engine: entweder WinLC, eine S7-CPU in Excel oder eine Slot-PLC wie die CPU 416-2 DP ISA
- Beispielprogramm (siehe Abschnitt 1.1)
- STEP 7 (zum Laden des Programms in die Control Engine und zum Einschalten der Eingangsbits im Peripheriebereich des Beispielprogramms)

Dieses Beispiel zeigt, wie Sie Ereignisse zum Aufrufen von Code einsetzen, der Ihre Excel-Felder aktualisiert. Ereignisse dienen dazu, sich ändernde Daten mit Code zu verknüpfen, der sich in einem VB-Arbeitsblatt befindet.

Hinweis

Sie können die Eingangsbits im Peripheriebereich des Beispielprogramms, das in der Control Engine ausgeführt wird, auch mit der Anwendung des Bedienpanel einschalten. Ausführliche Informationen zur Anwendung des Bedienpanel entnehmen Sie dem Abschnitt 1.2.

Erstellen einer Befehlsschaltfläche in Excel

Als ersten Schritt zum Erstellen einer Beispielanwendung in Excel erstellen Sie eine Befehlsschaltfläche. Gehen Sie folgendermaßen vor, um eine Befehlsschaltfläche zu erstellen:

1. Rufen Sie Excel auf. (Wenn Sie gefragt werden, ob Makros aktiviert oder deaktiviert werden sollen, geben Sie an, daß die Makros aktiviert werden sollen.)
2. Geben Sie in folgenden Feldern der Tabelle folgendes ein:
 - In Feld A1 tragen Sie ein: **ab0**
 - In Feld A2 tragen Sie ein: **ab1**
 - In Feld A3 tragen Sie ein: **ab2**
3. Wählen Sie den Menübefehl **Ansicht > Symbolleisten > Steuerelement-Toolbox**, um die Steuerelement-Toolbox aufzurufen.
4. Wählen Sie die Schaltfläche für den Entwurfsmodus in der Steuerelement-Toolbox, um die Excel-Tabelle in den Entwurfsmodus zu versetzen.
5. Fügen Sie eine Befehlsschaltfläche in die Tabelle ein, indem Sie in der Steuerelement-Toolbox auf das Symbol "Command Button" klicken und anschließend mit der linken Maustaste in einen freien Bereich in der Tabelle klicken.
6. Verschieben Sie die Befehlsschaltfläche, wenn Sie möchten, und ändern Sie auch die Größe nach Ihren Anforderungen.

Konfigurieren der Befehlsschaltfläche mit dem Visual Basic-Editor

Nachdem Sie die Befehlsschaltfläche erstellt haben, konfigurieren Sie die Schaltfläche mit dem Visual Basic-Editor in Excel, so daß mit der Schaltfläche das Programm gestartet und gestoppt werden kann.

Gehen Sie folgendermaßen vor, um die Befehlsschaltfläche zu konfigurieren:

1. Markieren Sie die Befehlsschaltfläche (CommandButton1).
2. Wählen Sie den Menübefehl **Extras > Makro > Visual Basic-Editor**, um den Visual Basic-Editor aufzurufen.
3. Im Dialogfeld "Eigenschaften" wählen Sie die Eigenschaft Caption für CommandButton1 und geben folgenden Text ein:
Zählen
4. Rufen Sie das Dialogfeld für den Code mit dem Menübefehl **Ansicht > Code** auf. Wählen Sie "CommandButton1" in dem aufklappbaren Listenfeld des Objekts aus.
Geben Sie folgenden Code für das Ereignis CommandButton1_Click() ein:
UserForm1.show
5. Schließen Sie das Dialogfeld für den Code für CommandButton1.

Erstellen eines SIMATIC Data Control

1. Erstellen Sie ein neues Anwenderformular mit dem Menübefehl **Einfügen > UserForm**.
2. Klicken Sie im Fenster "Steuerelement-Toolbox" mit der rechten Maustaste, um das Popup-Menü aufzurufen, und wählen Sie den Befehl **Steuerelemente...** (Zum Anzeigen des Fensters "Toolbox" wählen Sie den Menübefehl **Ansicht > Steuerelement-Toolbox**.)
3. Blättern Sie durch die Liste der Controls und wählen Sie das Data Control aus (indem Sie das Kontrollkästchen aktivieren). Wählen Sie "OK", um das Data Control in die Steuerelement-Toolbox aufzunehmen.
4. Wählen Sie das Symbol "Data Control" im Fenster "Steuerelement-Toolbox" und fügen Sie ein Data Control in das Fenster "UserForm1" ein.
5. Markieren Sie das Data Control und klicken Sie mit der rechten Maustaste, um das Popup-Menü aufzurufen. Im Popup-Menü wählen Sie den Befehl **Eigenschaften**, um die Eigenschaften des Data Control (S7Data1) im Fenster "Eigenschaften" anzuzeigen.
6. Im Fenster "Eigenschaften" für S7Data1 wählen Sie das Eigenschaftsfeld "(Benutzerdefiniert)" und klicken dann auf die Schaltfläche zum Erweitern des Fensters, damit das Dialogfeld "Eigenschaften" des Data Control angezeigt wird.

Ergänzen von Ereignissen für das Data Control

1. Im Dialogfeld "Eigenschaften" des Data Control öffnen Sie das Register "Events". In der Liste unter "Keys" wählen Sie S7Data1.
2. Wählen Sie die Schaltfläche "Add", um einen neuen Ereignisschlüssel aufzunehmen (siehe Bild 1-15). Im Dialogfeld "Add" tragen Sie **AB0** in das Feld "Add a new key" ein.

Klicken Sie auf die Schaltfläche "OK", damit der Ereignisschlüssel für das S7Data1 Control aufgenommen wird.

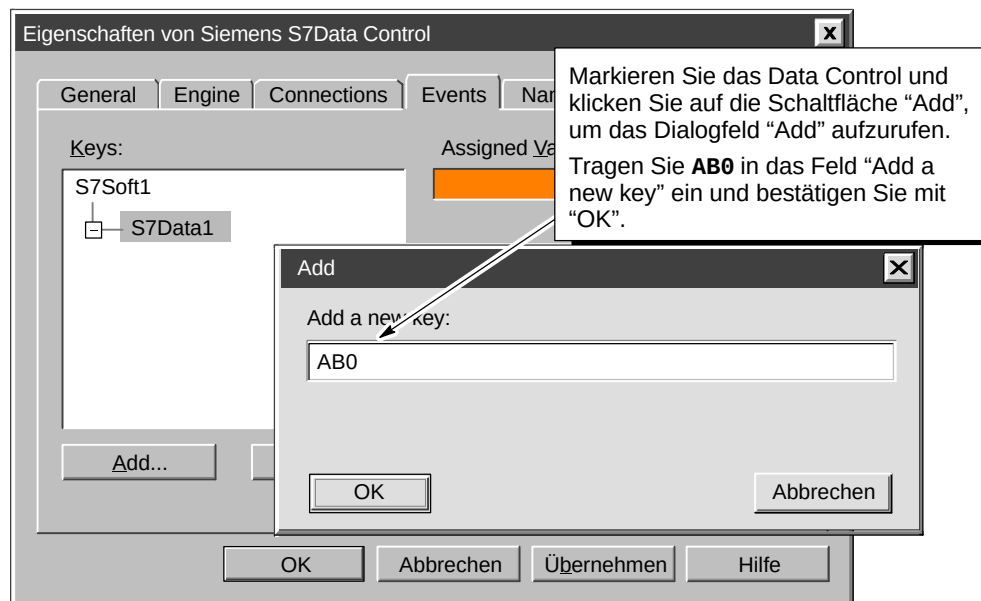


Bild 1-15 Ergänzen eines Ereignisses für das Data Control

3. Tragen Sie im Dialogfeld "Eigenschaften" die Adresse **AB0** in das Feld "Assigned Variable" ein (siehe Bild 1-16).
4. Wählen Sie die Schaltfläche "Übernehmen", um die zugewiesene Variable einzugeben. Daraufhin wird der Ereignisschlüssel "AB0" in Fettdruck unter S7Data1 angezeigt.
5. Geben Sie neue Ereignisschlüssel für AB1 (Adresse **AB1**) und AB2 (Adresse **AB2**) ein, indem Sie erneut S7Data1 anwählen und die Schritte 2. und 3. wiederholen.

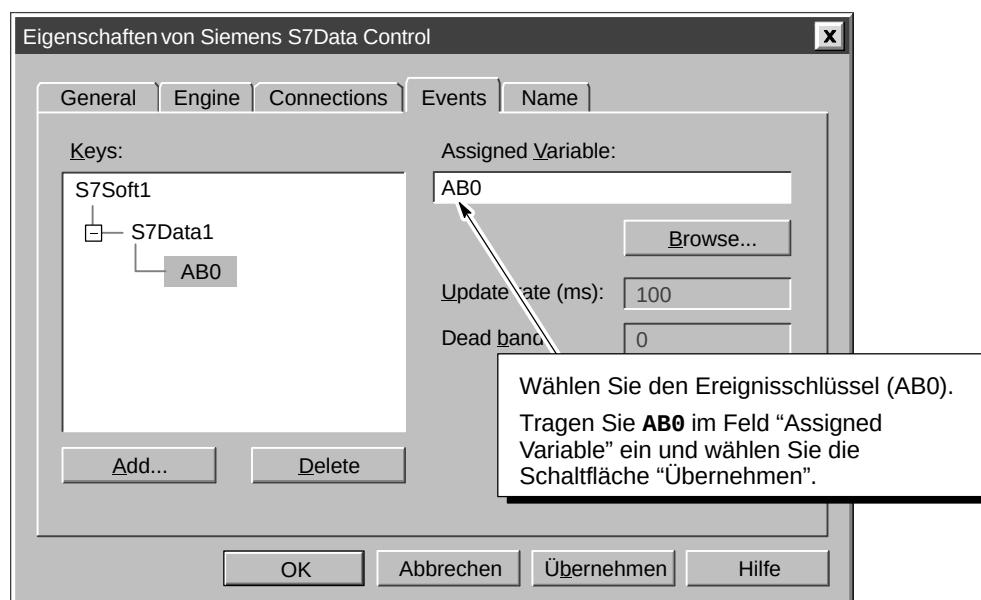


Bild 1-16 Zuordnen einer Variable zu einem Ereignisschlüssel

Konfigurieren der Control Engine für das Data Control

1. Öffnen Sie im Dialogfeld "Eigenschaften" für das Data Control das Register "Engine", um die Control Engine einzurichten.
2. Wählen Sie die Option "Direct Connect" und geben Sie **WinLC**, eine S7-CPU in Excel oder **wsS7=3** (für eine Slot-PLC wie die CPU 416-2 DP ISA) als Control Engine ein. Wählen Sie die Schaltfläche "Übernehmen", um die Daten zu übernehmen und bestätigen Sie mit "OK", um das Dialogfeld zu schließen.

Eingeben eines Beispielprogramms für das Data Control

1. Wählen Sie das Data Control in UserForm1.
2. Wählen Sie den Menübefehl **Ansicht > Code**, um das Fenster mit dem Code für das Data Control anzuzeigen.
3. Wählen Sie unter den Objekten das S7Data1 im aufklappbaren Listenfeld aus.
4. Geben Sie für das Ereignis S7Data1_ValueChanged folgendes Programm ein:

```
Select Case Property  
    Case "AB0"  
        Worksheets("Tabelle1").Range("B1").Value = Value  
    Case "AB1"  
        Worksheets("'Tabelle1").Range("B2").Value = Value  
    Case "AB2"  
        Worksheets("Tabelle1").Range("B3").Value = Value  
End Select
```

5. Schließen Sie das Fenster für den Code des Data Control und schließen Sie UserForm1.

Ausführen des Beispielprogramms

1. Wählen Sie den Menübefehl **Datei > Schließen und zurück zu Microsoft Excel**, um in die Excel-Tabelle zurückzukehren.
2. Verlassen Sie die Betriebsart Design, indem Sie auf die entsprechende Schaltfläche in der Toolbox klicken.
3. Verbinden Sie die Excel-Tabelle mit der Control Engine, indem Sie auf die Schaltfläche "Zählen" klicken.
4. Starten und stoppen Sie das Beispielprogramm in der Control Engine mit dem Bedienpanel (siehe Abschnitt 1.2).

Hinweis

Wenn Sie Excel schließen oder die Excel-Menüs aktivieren möchten, müssen Sie zunächst UserForm1 schließen.

1.5 Arbeiten mit dem SoftContainer von SIMATIC Computing

SIMATIC Computing verfügt über einen einfachen OLE-Container (SoftContainer) zum Anzeigen und Ändern der Daten aus der Control Engine. Mit diesem Container können Sie die SIMATIC Controls schnell in ein Prozeß-Formular einfügen. (Ein Prozeß-Formular ist ein Dokument bzw. eine Datei des SoftContainers, das verschiedene Controls enthält.) Mit diesem Werkzeug kann kein Code geschrieben werden.

Damit Sie dieses Beispiel für ein Prozeß-Formular ausführen können, müssen Sie das Beispielpogramm (siehe Abschnitt 1.1) in die Control Engine geladen haben.

Einfügen eines SIMATIC Control in ein Prozeß-Formular

Zum Aufrufen der Software SIMATIC Computing wählen Sie im Windows Startmenü den Menübefehl **Simatic > PC Based Control > Windows SIMATIC Computing SoftContainer**. Der SoftContainer öffnet sich und zeigt ein leeres Prozeß-Formular (S7Soft1) an. Sie fügen die SIMATIC Controls in dieses Prozeß-Formular ein (siehe Bild 1-17).

1. Klicken Sie in der Funktionsleiste auf das Symbol für das Data Control. (Wenn Sie den Mauszeiger auf einem Symbol positionieren und eine Sekunde nicht bewegen, wird ein Tooltip mit einer Kurzbeschreibung der Schaltfläche angezeigt.)
2. Bewegen Sie den Zeiger in das geöffnete Prozeß-Formular. Sie werden feststellen, daß sich der Mauszeiger zu einem Fadenkreuz verwandelt, wenn Sie ihn im Prozeß-Formular bewegen.
3. Klicken Sie mit der linken Maustaste, um das Data Control einzufügen.

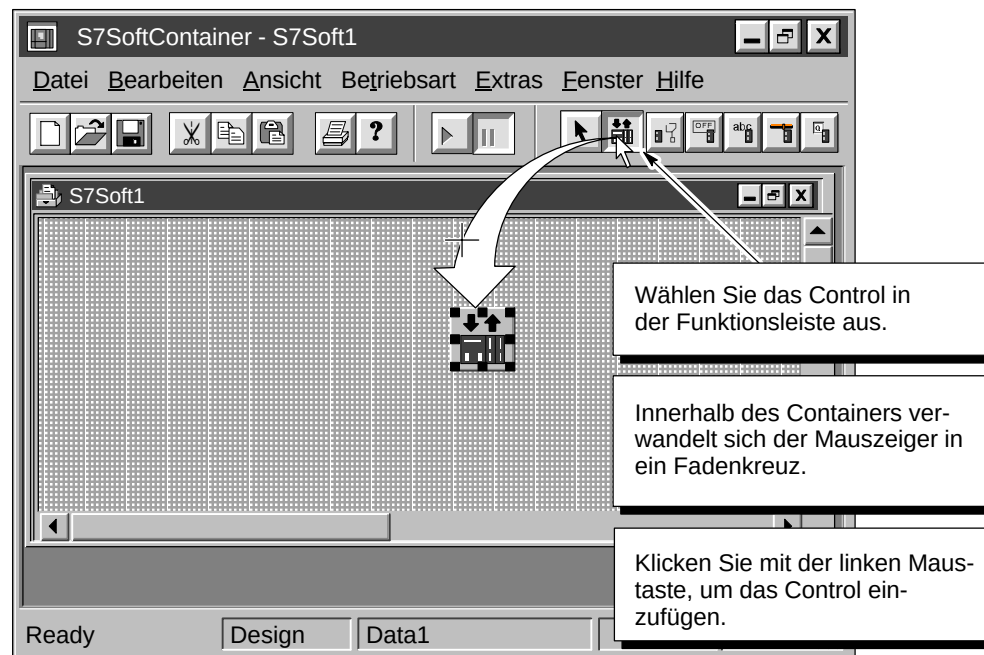


Bild 1-17 Einfügen eines SIMATIC Control in den WinAC SoftContainer

Wiederholen Sie diese Schritte, um drei Button Controls und drei Edit Controls einzufügen. (Ausführliche Informationen zum Einfügen von Controls in den SoftContainer entnehmen Sie dem Abschnitt 8.2.) Bild 1-18 zeigt ein Beispiel für die Anordnung der Controls im Prozeß-Formular (S7Soft1).

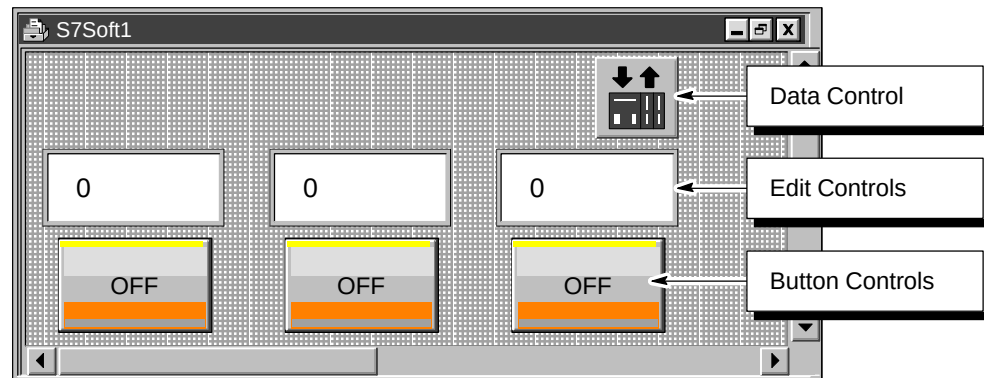


Bild 1-18 Erstellen eines Bedienpanels im SoftContainer

Konfigurieren der Eigenschaften der SIMATIC Controls

Im Dialogfeld "Eigenschaften" des Data Control verbinden Sie die anderen SIMATIC Controls mit der Control Engine.

Zum Zuordnen einer Variablen (Adresse im Speicher der Control Engine) zu einem SIMATIC Control markieren Sie das Data Control und klicken Sie mit der rechten Maustaste, um das Popup-Menü aufzublenzen. Im Popup-Menü wählen Sie den Befehl **Eigenschaften**, um das Dialogfeld "Eigenschaften" des Data Control anzuzeigen.

Konfigurieren der Control Engine für das Data Control

In diesem Beispiel wird davon ausgegangen, daß Sie eine Control Engine installiert haben. Informationen zum Anschließen von Control Engines entnehmen Sie dem Abschnitt 5.3.

Gehen Sie folgendermaßen vor, um das Data Control mit der Control Engine zu verbinden:

1. Im Dialogfeld "Eigenschaften" des Data Control öffnen Sie das Register "Engine".
2. Wählen Sie die Option "Direct Connect" und geben Sie entweder **WinLC** oder **wcS7=3** (für eine Slot-PLC wie die CPU 416-2 DP ISA) für die Control Engine ein.
3. Wählen Sie die Schaltfläche "Übernehmen", um diese Daten einzugeben.

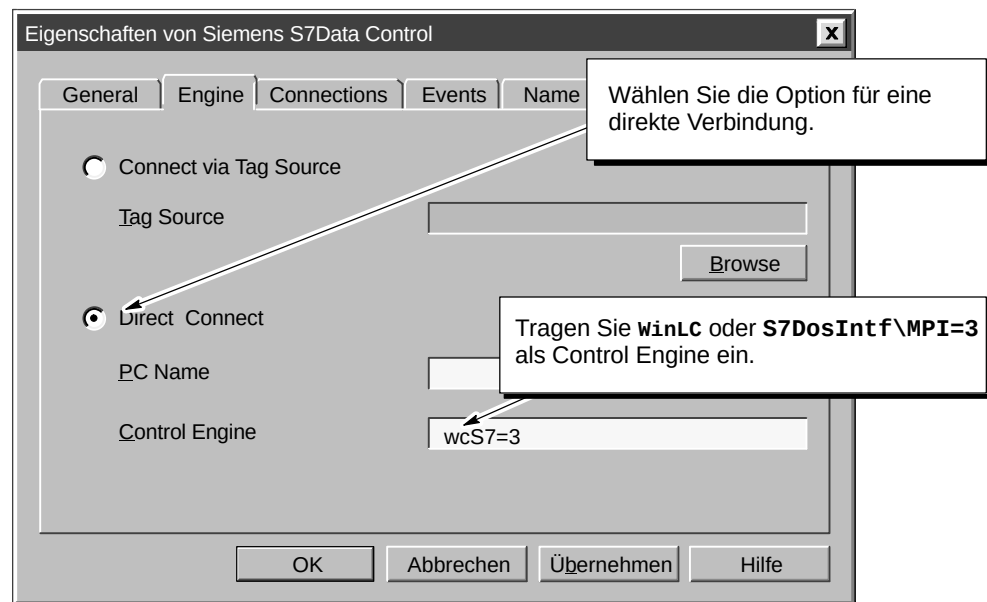


Bild 1-19 Verbinden mit der Control Engine (Beispiel: SoftContainer)

Zuordnen einer Variablen (Adresse im Speicher) zu einer Eigenschaft

Das Data Control stellt eine Verbindung zwischen den einzelnen SIMATIC Controls und der Control Engine her. Im Dialogfeld "Eigenschaften" des Data Control ordnen Sie den einzelnen Eigenschaften der Controls Variablen (Adressen im Speicher der Control Engine) zu.

Ordnen Sie die Variablen (Adressen im Speicher der Control Engine) zu den SIMATIC Controls wie in Tabelle 1-2 beschrieben zu.

Tabelle 1-2 Zuordnen von Beispielandressen zu den SIMATIC Controls

Control	Adresse	Beschreibung
Edit1	AB0	Ausgabewert des ersten Zählers
Edit2	AB1	Ausgabewert des zweiten Zählers
Edit3	AB2	Ausgabewert des dritten Zählers
Button1	E0.0	Freigabebit für den ersten Zähler
Button2	E0.1	Freigabebit für den zweiten Zähler
Button3	E0.2	Freigabebit für den dritten Zähler

Gehen Sie folgendermaßen vor, um die Eigenschaft Value des Button Control Button1 mit PE0.0 in der Control Engine zu verbinden:

1. Im Dialogfeld "Eigenschaften" des Data Control öffnen Sie das Register "Connections".
2. Klicken Sie auf das Symbol "+" neben SIMATIC Data1 (oder doppelklicken Sie auf SIMATIC Data1), um die Liste der Controls im Container anzuzeigen.
3. Klicken Sie auf das Symbol "+" neben Button1 (oder doppelklicken Sie auf Button1), um die Liste der Eigenschaften des Button Control anzuzeigen (siehe Bild 1-20).
4. Markieren Sie die Eigenschaft Value. Sie werden feststellen, daß das Feld "Assigned Variable" aktiviert wird, wenn Sie die Eigenschaft Value auswählen.
5. Geben Sie im Feld "Assigned Variable" die Adresse "E0.0" ein (siehe Bild 1-20). (Sie können sowohl Klein- als auch Großbuchstaben zum Eingeben der Adressen verwenden.)
6. Wählen Sie die Schaltfläche "Übernehmen", um diese Daten einzugeben.

Wiederholen Sie diese Vorgehensweise für die anderen Button Controls und die drei Edit Controls und geben Sie die in Tabelle 1-2 aufgeführten Variablen ein. Nachdem Sie die Verbindungen für alle Controls eingerichtet haben, bestätigen Sie Ihre Eingaben mit "OK" und schließen das Dialogfeld "Eigenschaften".

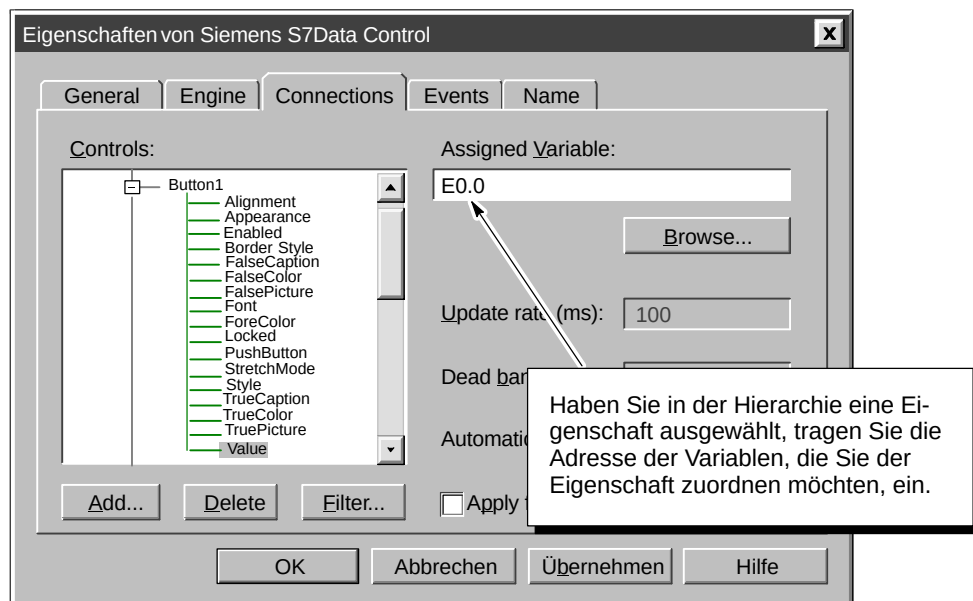


Bild 1-20 Zuordnen der Eigenschaft Value zu einer Variablen

Konfigurieren des Edit Control für Binärdaten

Das Edit Control kann Daten in verschiedenen Formaten anzeigen. In diesem Beispiel richten Sie die Edit Controls so ein, daß die Datenbytes (AB0, AB1 und AB2) im Dezimalformat angezeigt werden.

Hinweis

Das Feld "Data Type" des Edit Control legt die Größe der Daten, die angezeigt werden sollen, fest.

Gehen Sie folgendermaßen vor, um das Edit Control zu konfigurieren:

1. Markieren Sie das Edit Control (Edit1) und klicken Sie mit der rechten Maustaste, um das Popup-Menü aufzublenzen. Im Popup-Menü wählen Sie den Befehl **Eigenschaften**, um das Dialogfeld "Eigenschaften" des Edit Control anzuzeigen.
2. Klicken Sie auf den Pfeil neben dem Feld "Data Format", um das aufklappbare Listenfeld aufzublenzen
3. Suchen Sie den Eintrag für dezimal und klicken Sie auf "2 - wDecimal", um den Wert im Binärformat (0 oder 1) anzuzeigen. (siehe Bild 1-21).
4. Wählen Sie die Schaltfläche "Übernehmen", um die Daten zu übernehmen und bestätigen Sie mit "OK", um das Dialogfeld "Eigenschaften" zu schließen.

Wiederholen Sie diese Vorgehensweise für die anderen Edit Controls (Edit2 und Edit3).

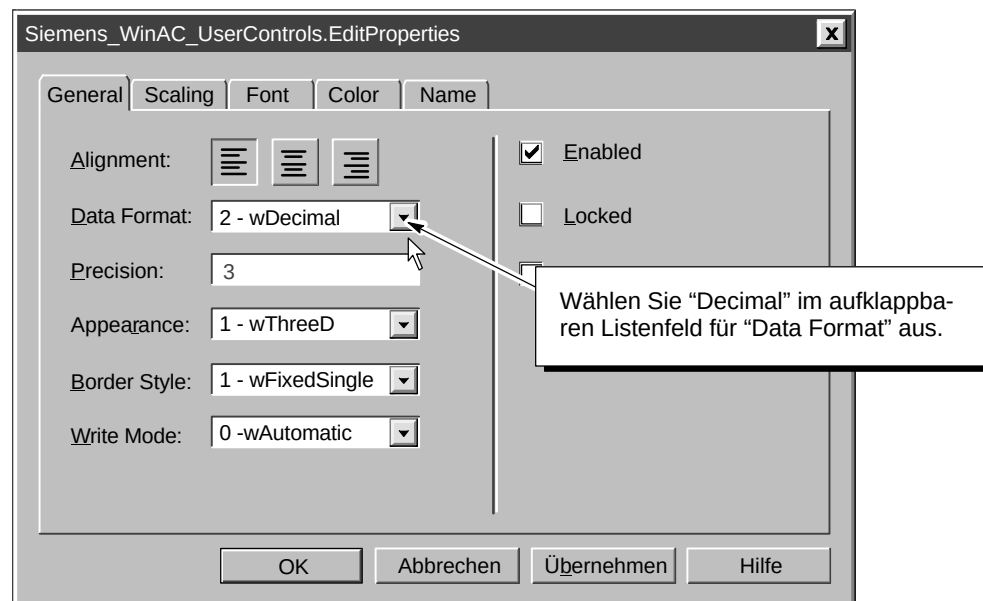


Bild 1-21 Konfigurieren der Anzeigeeigenschaften des Edit Control

Verbinden der SIMATIC Controls mit der Control Engine

Ist die Control Engine nicht aktiv, können Ihre Controls keinen Prozeß überwachen. Wenn Sie mit den Controls Daten überwachen oder ändern möchten, muß die Control Engine aktiviert sein.

Gehen Sie folgendermaßen vor, um die Controls im Container mit der Control Engine zu verbinden:

1. Klicken Sie auf das Symbol "RUN" (oder wählen Sie den Menübefehl **Betriebsart > RUN**), um den Container von der Betriebsart Design in die Betriebsart RUN umzuschalten (siehe Bild 1-22).
2. Klicken Sie auf die Button Controls, um die Zähler im Beispielprogramm zu starten (bzw. zu stoppen). Sie werden feststellen, daß sich der im entsprechenden Edit Control angezeigte Wert ändert, wenn sich der Zustand des Button Control ändert.
3. Klicken Sie auf das Symbol "Design" (oder wählen Sie den Menübefehl **Betriebsart > Design**), um die Betriebsart des Containers von RUN in Design zu schalten (dadurch werden die Controls von der Control Engine getrennt).

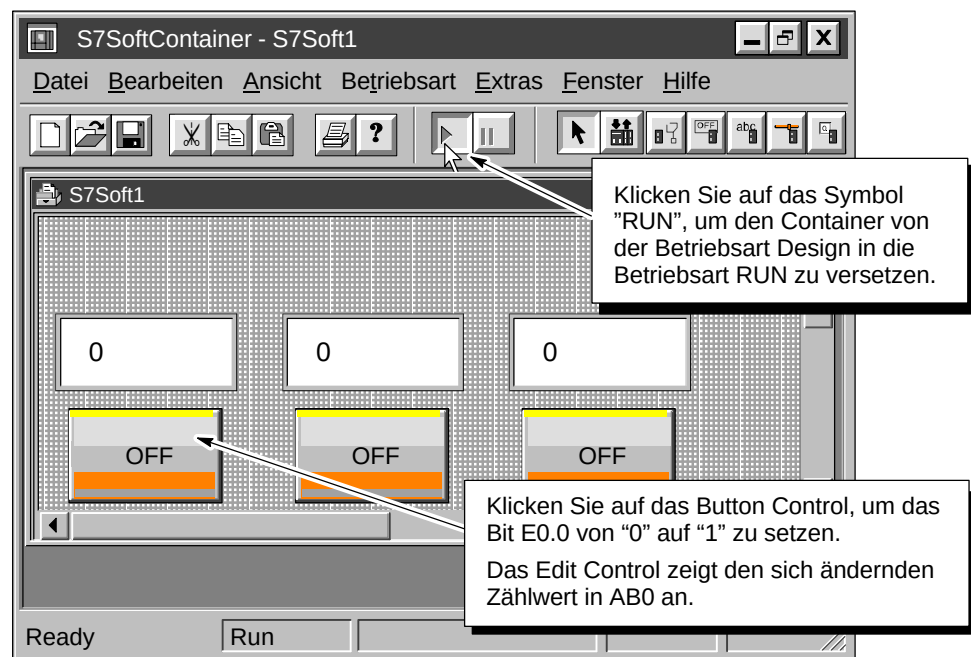


Bild 1-22 Versetzen des Containers in die Betriebsart RUN

Kapitelübersicht

Das Softwarepaket SIMATIC Computing bietet anderen Software-Anwendungen die Möglichkeit, auf Prozeßdaten aus Ihrer Anwendung zuzugreifen. Die Software SIMATIC Computing verfügt über ActiveX Controls, die in jede Anwendung eingefügt werden können, bei der es sich um einen Container von ActiveX Controls handelt, z.B. Visual Basic oder Visual C++.

Die SIMATIC Controls in SIMATIC Computing sind mit anderen Containern von anderen Herstellern getestet worden, doch kann es sein, daß die Funktionsweise einiger Fremd-Container von der in diesem Handbuch beschriebenen Funktionsweise abweicht. Richtlinien zu Fremd-Containern sowie zur Verwendung von anwenderspezifischen ActiveX Controls mit dem Data Control entnehmen Sie dem Anhang F.



Warnung

Wenn Sie den Wert, der in einem ActiveX Control angezeigt wird, ändern, ganz gleich, ob Sie die Änderung in SIMATIC Computing oder in einer Fremd-Software vornehmen, ändern Sie den Wert im tatsächlichen Prozeß.

Durch das Ändern von Prozeßdaten können Sie unvorhersehbare Reaktionen im Prozeß auslösen, die zu Tod, schwerer Körperverletzung und/oder Sachschaden führen können.

Gehen Sie vorsichtig vor, ändern Sie keine Daten und gewähren Sie unberechtigten Personen keinen Zugriff auf Daten, die unerwartete Funktionsweise der gesteuerten Geräte hervorrufen können. Installieren Sie einen physikalischen NOT-AUS-Schaltkreis für die Maschine bzw. den Prozeß.

Kapitel	Beschreibung	Seite
2.1	Produktübersicht	2-2
2.2	Zugreifen auf Prozeßdaten mittels ActiveX Control	2-4
2.3	Herstellen einer Verbindung zum Prozeß mit dem OPC-Server in WinAC	2-6

2.1 Produktübersicht

Mit der Software SIMATIC Computing können Sie auf die Control Engine in Ihrem Prozeß zugreifen, um Prozeßdaten zu überwachen und zu ändern. Bild 2-1 zeigt, wie Sie die Software SIMATIC Computing mit mehreren Control Engines, z.B. dem Windows Logic Controller (WinLC), einer Slot-PLC wie der CPU 416-2 DP ISA oder mit S7-Systemen einsetzen können. Je nach der Kommunikationskarte in Ihrem PC können Sie auf die S7-Automatisierungssysteme über ein MPI-, PROFIBUS-DP- oder Industrial Ethernet-Netz zugreifen.

In der Software SIMATIC Computing können Sie – anstelle von absoluten Adressen – symbolische Namen einsetzen, um auf Speicherbereiche oder Control Engines zuzugreifen. Diese Symbole werden in einer Symboldatei gespeichert, die automatisch aus der Symboltabelle des Projekts in STEP 7 angelegt wird.

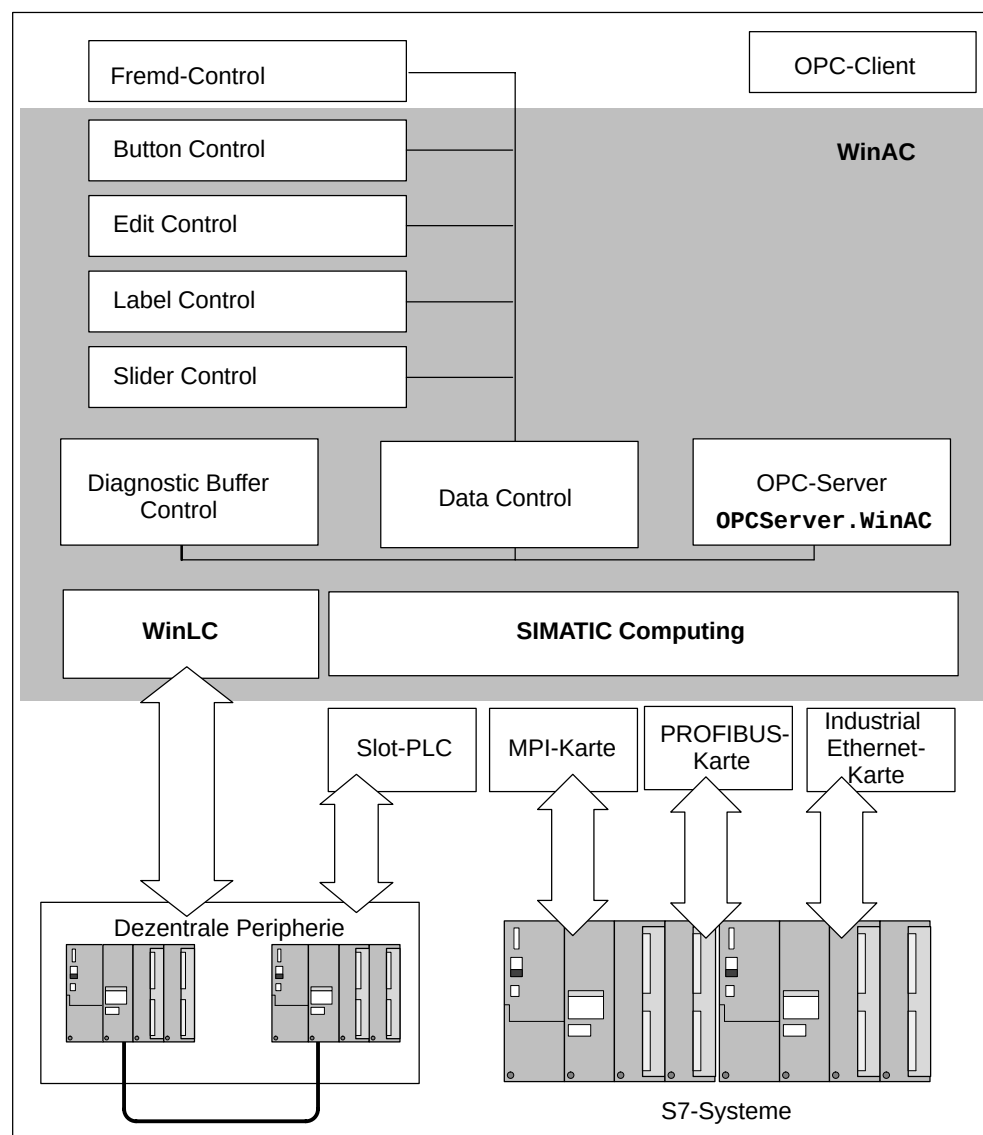


Bild 2-1 Zugreifen auf Prozeßdaten mit SIMATIC Computing

Wie Bild 2-1 zeigt, gibt es in SIMATIC Computing verschiedene Methoden, um auf Prozeßdaten zuzugreifen:

- SIMATIC Computing bietet über das Data Control Standard-ActiveX-Controls, die auf Prozeßdaten zugreifen. Sie können sie mit dem SIMATIC Computing Container, der im Lieferumfang der Software enthalten ist, verwenden, oder Sie fügen diese Controls in Container anderer Softwarepakete ein.
- SIMATIC Computing bietet einen Diagnosepuffer für die S7-Automatisierungssysteme. Dieser Diagnosepuffer ist ein Ringpuffer, der Einträge enthält, die vom Betriebssystem der S7-Automatisierungssysteme geschrieben werden. Jeder Eintrag enthält Informationen zu einem bestimmten Diagnoseereignis. Das DBuffer Control gewährt Ihrem Programm Zugriff auf den Diagnosepuffer und es zeigt die Ereignisse an.
- SIMATIC Computing verfügt über einen OPC-Server (OPC = OLE for Process Control), über den andere OPC-Anwendungen auf die Daten in dem gesteuerten Gerät zugreifen können. SIMATIC Computing enthält keine OPC-Client-Anwendung.

Der OPC-Server basiert auf der OLE/COM-Technologie von Microsoft. Ausführliche Informationen zu OPC entnehmen Sie folgendem OPC-Dokument: *OLE for Process Control Data Access Standard, Version 2.0*, herausgegeben von der OPC Foundation.

Systemanforderungen

Zum Installieren der Software SIMATIC Computing empfehlen wir Ihnen folgende Voraussetzungen für Ihren Computer:

- Einen Personal Computer (PC) mit:
 - Pentium-Prozessor mit 166 MHz oder schneller (empfohlen)
 - 64 MB RAM-Speicher
 - Microsoft Windows NT Version 4.0 (oder höher) mit Service Pack 3
- Einen Farbmonitor, eine Tastatur und eine Maus (oder ein anderes Zeigegerät), die von Microsoft Windows NT unterstützt werden
- Eine Festplatte mit 20 MB freiem Speicherplatz
- Mindestens 1 MB freien Speicherplatz auf Laufwerk C für das Setup-Programm (die Setup-Dateien werden nach erfolgreicher Installation gelöscht)

Das Produkt wurde sogar auf einem langsamen Rechner mit einem Prozessor 486, 66 MHz und 24 MB RAM-Speicher auf der Plattform Windows NT erfolgreich getestet. SIMATIC Computing wurde auch erfolgreich auf einem leistungsstarken PC mit dualen Pentium-Prozessor getestet.

2.2 Zugreifen auf Prozeßdaten mittels ActiveX Control

SIMATIC Computing bietet über das Data Control Zugriff auf die Prozeßdaten, die von einer Control Engine wie WinLC (Windows Logic Controller) gesteuert werden. Sie können mit den Standard-SIMATIC-Controls, die im Lieferumfang der Software SIMATIC Computing enthalten sind (siehe Tabelle 2-1), arbeiten, oder Sie können beliebige andere ActiveX Controls mit dem Data Control verbinden.

In SIMATIC Computing können Sie keine Daten in Zeiten schreiben. Sie können die Zeitwerte nur lesen.

Tabelle 2-1 Standard-Controls in SIMATIC Computing

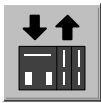
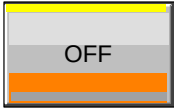
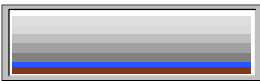
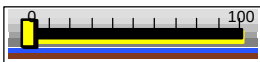
Control	Darstellung	Beschreibung
Data		Stellt die Verbindung zur Control Engine (z.B. WinLC) her. Ohne das Data Control hat keines der anderen Controls Zugriff auf die Prozeßdaten.
Button		Bietet Zugriff auf einzelne Bits im Speicher der Control Engine. Das Button Control greift nur im Bitformat zu und verfügt über zwei Werte: • Off = 0 (voreingestellte Farbe: rot) • On = 1 (voreingestellte Farbe: grün) Wenn Sie den Zustand des Button Control ändern, wird auch der Zustand der Variablen in Ihrem Prozeß, die dem Control zugeordnet ist, geändert. Richten Sie für das Button Control Schreibschutz ein, wird es wie eine Lampe oder eine LED betrieben. Richten Sie das Button Control als Taste ein, wird es wie ein Schalter betrieben.
Edit		Bietet Zugriff auf die Adressen im Speicher der Control Engine. Sie können im Byte-, Wort- oder Doppelwortformat zugreifen und Sie können einzelne Bits dieser Daten ändern. Wenn Sie im Number Control einen neuen Wert eingeben, werden die Daten in der Control Engine geändert.
Label		Mit dem Label Control können Sie eine Konstante als Zeichenfolge anzeigen. Sie können die Eigenschaft Caption des Label Control auch mit einem beliebigen Prozeßwert verbinden. Der Prozeßwert wird in eine Zeichenfolge umgewandelt und angezeigt.
Slider		Bietet Zugriff auf die Adressen im Speicher der Control Engine. Sie können im Byte-, Wort- oder Doppelwortformat zugreifen. Wenn Sie im Slider Control einen Wert anpassen, werden die Daten in der Control Engine geändert.

Tabelle 2-1 Standard-Controls in SIMATIC Computing, Fortsetzung

Control	Darstellung	Beschreibung
DBuffer (S7-Diagnosepuffer)		Zeigt den Diagnosepuffer des Controller an. Das DBuffer Control ist direkt mit dem Controller verbunden: es muß nicht über das Data Control verbunden werden.



Vorsicht

Wenn Sie die Zeitfunktion oder Haltepunkte in Visual Basic mit SIMATIC Computing verwenden, kann dies zu Fehlern führen, die evt. Ihren Computer bzw. die Anwendung zum Absturz bringen bzw. verriegeln können. Je nach Konfiguration kann die Anwendung dadurch die Kommunikation mit der Control Engine beenden. Dies kann unvorhersehbare Reaktionen im Prozeß auslösen, die zu Tod, schwerer Körperverletzung und/oder Sachschaden führen können.

Installieren Sie einen physikalischen NOT-AUS-Schaltkreis für die Maschine bzw. den Prozeß.

Timer in VB: Bei der Timer-Funktion in Visual Basic Version 5 kann ein Timer den in der gleichen Einweg-Bearbeitung befindlichen Code unterbrechen, wodurch Fehler mit schwerwiegenden Folgen auftreten können. Beachten Sie folgende Richtlinien, wenn Sie VB-Timer mit SIMATIC Computing einsetzen:

- Deaktivieren Sie die Timer immer im Unterprogramm Form_Unload. Andernfalls kann ein Timer ein Ereignis auslösen, während das VB-Programm geschlossen wird. Diese Bedingung kann einen Absturz bzw. eine Verriegelung Ihres Computers oder der Anwendung herbeiführen. Es kann auch sein, daß die Anwendung unsichtbar weiterläuft.
- Wenn Sie den Timer im Unterprogramm Form_Load starten, kann es sein, daß das Zeitereignis auftritt, bevor die anderen Objekte instanziiert wurden. Um sicherzustellen, daß die Objekte korrekt instanziiert werden, starten Sie einen Timer im Unterprogramm Form_Load immer mit einem großen Intervall (z.B. 1 oder 2 Sekunden). Nachfolgende Timer-Intervalle können kürzer sein.

2.3 Herstellen einer Verbindung zum Prozeß mit dem OPC-Server in WinAC

OLE for Process Control (OPC) stellt einen Standardmechanismus zum Kommunizieren mit einer Vielzahl von Datenquellen zur Verfügung. Hierbei spielt es keine Rolle, ob es sich bei diesen Quellen um Maschinen in Ihrer Fabrik oder um eine Datenbank in Ihrer Schaltzentrale handelt. Mit Hilfe des OPC-Servers, der im Lieferumfang der Software SIMATIC Computing enthalten ist, können Sie mit der Control Engine (z.B. dem WinLC Controller) kommunizieren und so auf die Prozeßdaten zugreifen. SIMATIC Computing enthält einen OPC-Server, über den jede OPC-Client-Anwendung auf Daten in der Control Engine zugreifen kann. SIMATIC Computing verfügt über keine OPC-Client-Anwendung.

SIMATIC Computing implementiert nur die erforderlichen Schnittstellen, die in Version 2.0 der Dokumentation der OPC Foundation definiert sind. Die in dieser Dokumentation als "custom" definierten Schnittstellen werden gegebenenfalls zu einem späteren Zeitpunkt implementiert.

OPC basiert auf der OLE/COM-Technologie von Microsoft. Ausführliche Informationen zu OPC entnehmen Sie der OPC-Dokumentation *OLE for Process Control Data Access Standard, Version 2.0*, die von der OPC Foundation herausgegeben wurde.

Verbinden von SIMATIC Computing mit Client-Applikationen

Über OPC können Sie auf Daten aus der Fertigungshalle zugreifen und diese Daten in Ihre bestehenden Systeme zur Geschäftsabwicklung übernehmen. Sie können handelsübliche Werkzeuge (z.B. SCADA-Pakete, Datenbanken, Tabellenkalkulationen) einsetzen, um ein System zu erstellen, das Ihren Anforderungen entspricht. Wie Bild 2-2 zeigt, verfügt OPC über eine offene und effektive Kommunikationsarchitektur, die den Schwerpunkt auf den Datenzugriff und nicht auf die Datentypen legt.

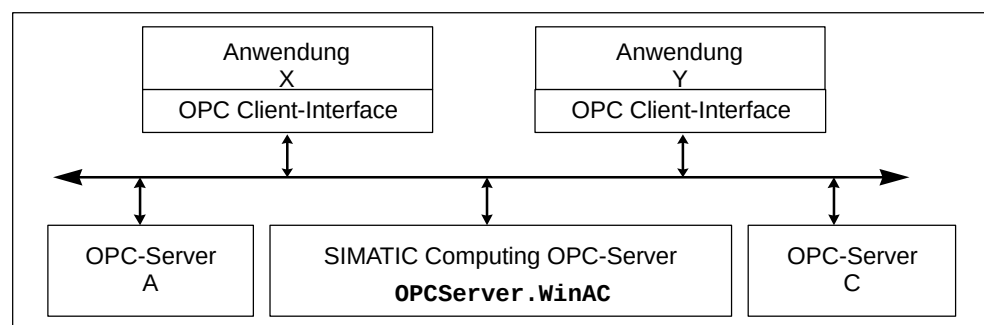


Bild 2-2 Anwendungen, die mit vielen OPC-Servern arbeiten

Der OPC-Client wird über SIMATIC Computing mit dem OPC-Server verbunden. Durch diese Verbindung können Sie OPC-Gruppen erstellen und bearbeiten. OPC-Gruppen strukturieren die Daten, auf die zugegriffen wird. Sie können eine Gruppe als Einheit aktivieren oder deaktivieren, oder Sie können sich in die Liste einer Gruppe von Objekten eintragen, damit Ihnen mitgeteilt wird, wenn sich die Daten ändern. (Eine Gruppe ist eine Sammlung von Elementen wie MB0.) Bild 2-3 zeigt die Verbindung zwischen der OPC-Client-Anwendung und den Prozeßdaten über WinAC.

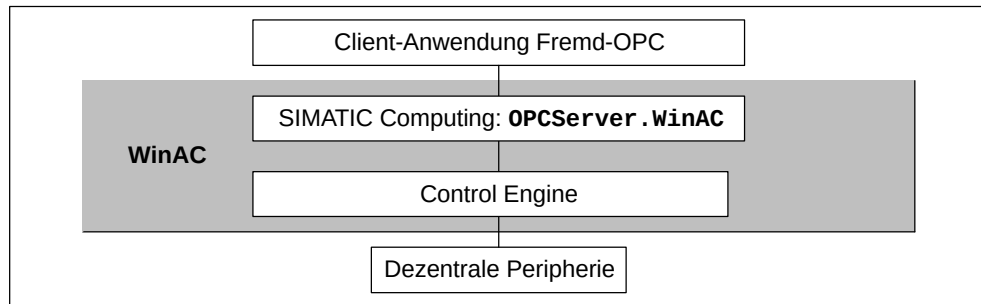


Bild 2-3 Zugreifen auf Prozeßdaten über den OPC-Server in WinAC

Wenn Sie auf den OPC-Server und seinen Inhalt zugreifen möchten, müssen Sie Ihrem OPC-Client den Namen (ProgID bzw. Programmierkennung) des Servers mitteilen: Der Name des in SIMATIC Computing vorhandenen OPC-Servers lautet: OPCServer.WinAC

Ausführliche Informationen zum OPC-Server in SIMATIC Computing entnehmen Sie dem Handbuch *OPC Server Interface*.

Einrichten der Software SIMATIC Computing

3

Kapitelübersicht

Dieses Kapitel enthält folgende Informationen:

- Abschnitt 3.1 führt die Anforderungen zum Installieren und Betreiben der Software SIMATIC Computing auf.
- In Abschnitt 3.2 wird beschrieben, wie Sie die Autorisierung installieren.
- In Abschnitt 3.3 wird beschrieben, wie Sie die Software SIMATIC Computing installieren und deinstallieren.
- In Abschnitt 3.4 wird beschrieben, wie Sie die Software SIMATIC Computing mittels der Anwendung "PG/PC-Schnittstelle einstellen" mit einer Slot-PLC oder einer Kommunikationskarte verbinden.

Kapitel	Beschreibung	Seite
3.1	Übersicht	3-2
3.2	Autorisierung	3-3
3.3	Installieren und Deinstallieren der Software SIMATIC Computing	3-5
3.4	Verbinden von SIMATIC Computing mit einer Slot-PLC oder einer Kommunikationskarte	3-7

3.1 Übersicht

Die Software SIMATIC Computing verfügt über ActiveX Controls, die sich so einrichten lassen, daß Sie ganz spezifisch Ihren Prozeß überwachen können. In SIMATIC Computing können Sie neben S7-Controls auch ActiveX Controls von anderen Herstellern zum Überwachen und Ändern von Prozeßdaten einsetzen.

Richten Sie sich bezüglich der Autorisierung Ihrer Software SIMATIC Computing nach den folgenden Richtlinien.

- Für WinAC Basis verwenden Sie die Autorisierung für WinAC.
- Für SIMATIC Net verwenden Sie die Autorisierung für SIMATIC Net.
- Zum Aufrüsten von WinAC Pro mit SIMATIC Computing 3.0 Standalone verwenden Sie die Autorisierung für SIMATIC Computing.
- Für die Standalone-Version verwenden Sie die Autorisierung für SIMATIC Computing.

Hinweis

Damit Sie SIMATIC Computing auf einem anderen PC als WinLC betreiben können, benötigen Sie die Standalone-Version von SIMATIC Computing.

3.2 Autorisierung

Für die Nutzung von SIMATIC Computing benötigen Sie eine produktspezifische Autorisierung (Nutzungsberechtigung). Die Software ist so kopiergeschützt und nur benutzbar, wenn auf der Festplatte des betreffenden Computers die für das Programm oder Softwarepaket erforderliche Autorisierung erkannt wird.

Hinweis

Wenn Sie die Autorisierung deinstallieren, können Sie SIMATIC Computing weiter nutzen. Es erscheint jedoch alle sechs Minuten eine Warnung, die Sie darüber informiert, daß Sie ohne Autorisierung arbeiten.

Autorisierungsdiskette

Die Software enthält eine Autorisierungsdiskette. Diese Diskette enthält die Autorisierung und das Programm (AUTHORSW), mit dem Sie die Autorisierung anzeigen, installieren und deinstallieren können.

Jede SIMATIC Automatisierungssoftware enthält eine eigene Autorisierungsdiskette. Die Autorisierung für ein Produkt installieren Sie als Teil der Software-Installation.



Vorsicht

Wird die Autorisierung für SIMATIC Computing fehlerhaft übertragen bzw. deinstalliert, kann sie dabei unwiderbringlich verloren gehen.

Die Liesmich-Datei auf der Autorisierungsdiskette enthält Richtlinien zum Installieren, Übertragen und Deinstallieren der Autorisierung für SIMATIC Computing. Wenn Sie diese Richtlinien nicht beachten, besteht die Gefahr, daß die Autorisierung für SIMATIC Computing unwiderbringlich verloren geht. Ohne Autorisierung können Sie keine Programme mehr in SIMATIC Computing laden und die bereits in SIMATIC Computing geladenen Programme nicht mehr ändern.

Lesen Sie die Hinweise in der Liesmich-Datei auf der Autorisierungsdiskette und beachten Sie die Richtlinien zum Übertragen und Deinstallieren von Autorisierungen.

Installieren der Autorisierung

Bei der Erstinstallation Ihrer Software werden Sie durch eine Meldung aufgefordert, die Autorisierung zu installieren. Gehen Sie folgendermaßen vor, um die Autorisierung für die Software SIMATIC Computing zu installieren:

1. Legen Sie die Autorisierungsdiskette in das Laufwerk ein, wenn die entsprechende Aufforderung angezeigt wird.
2. Bestätigen Sie die Eingabeaufforderung.

Die Autorisierung wird auf die Festplatte (C:) übertragen, d.h. Ihr Rechner registriert, daß Sie eine Berechtigung haben.

Hinweis

Geben Sie bei der Installation der Autorisierung für SIMATIC Computing als Ziel-Laufwerk immer Laufwerk C: an.

Wenn Sie SIMATIC Computing starten und keine Autorisierung vorhanden ist, erhalten Sie eine entsprechende Meldung. Um die Autorisierung nachträglich vorzunehmen, rufen Sie das Programm AUTHORSW auf der Autorisierungsdiskette auf. Mit diesem Programm können Sie Autorisierungen anzeigen, installieren und deinstallieren.

Deinstallieren der Autorisierung

Wird eine erneute Autorisierung notwendig, z.B. wenn Sie das Laufwerk, auf dem sich die Berechtigung befindet, neu formatieren wollen, so müssen Sie vorher die Autorisierung "retten". Hierzu benötigen Sie die Original-Autorisierungsdiskette.

Gehen Sie folgendermaßen vor, um die Autorisierung wieder zurück auf die Autorisierungsdiskette zu übertragen:

1. Legen Sie die Original-Autorisierungsdiskette in das Diskettenlaufwerk ein.
2. Rufen Sie das Programm AUTHORSW.EXE von der Autorisierungsdiskette auf.
3. Wählen Sie aus der Liste aller Autorisierungen auf Laufwerk C: die Autorisierung aus, die Sie deinstallieren möchten.
4. Wählen Sie den Menübefehl **Autorisierung > Übertragen....**
5. Geben Sie im Dialogfeld das Ziel-Diskettenlaufwerk an, auf das die Autorisierung übertragen werden soll, und bestätigen Sie.
6. Danach wird wieder das Dialogfeld mit der Liste der restlichen Autorisierungen auf dem Laufwerk angezeigt. Schließen Sie das Programm AUTHORSW, wenn Sie keine weiteren Autorisierungen deinstallieren möchten.

Sie können diese Diskette dann erneut zum Autorisieren verwenden. Zum Deinstallieren von vorhandenen Autorisierungen benötigen Sie die Autorisierungsdiskette. Wenn Sie SIMATIC Computing komplett deinstallieren möchten, müssen Sie die DP-Autorisierung deinstallieren.

Tritt auf Ihrer Festplatte ein Defekt auf, bevor Sie die Berechtigung retten konnten, wenden Sie sich bitte an Ihre zuständige Siemens-Vertretung.

3.3 Installieren und Deinstallieren der Software SIMATIC Computing

Die Software SIMATIC Computing umfaßt ein Setup-Programm, das die Installation automatisch durchführt. Eingabeaufforderungen auf dem Bildschirm führen Sie Schritt für Schritt durch den gesamten Installationsvorgang.

Hinweis

Zum Installieren der Software SIMATIC Computing benötigen Sie Administratorrechte ("ADMIN").

Aufrufen des Installationsprogramms

Das Installationsprogramm führt Sie schrittweise durch den Installationsvorgang. Sie können jederzeit zum nächsten oder auch vorherigen Schritt weiter- bzw. zurückschalten. Zum Aufrufen des Installationsprogramms gehen Sie folgendermaßen vor:

1. Legen Sie die CD-ROM in das CD-Laufwerk Ihres Computers.
2. Öffnen Sie über das Windows NT Startmenü (Menübefehl **Start > Ausführen**) das Dialogfeld "Ausführen".
3. Klicken Sie im Dialogfeld "Ausführen" auf die Schaltfläche "Durchsuchen" und wählen Sie das Installationsprogramm (Setup.exe) auf der CD-ROM.
4. Wählen Sie die Schaltfläche "Öffnen", um das Programm Setup.exe in das Dialogfeld "Ausführen" zu übernehmen.
5. Bestätigen Sie mit "OK", um das Installationsprogramm zu starten.
6. Folgen Sie den Anweisungen, die Ihnen das Installationsprogramm anzeigt.
Wenn Sie nur die Software SIMATIC Computing installieren möchten, deaktivieren Sie die anderen Komponenten von WinAC.
7. Werden Sie von der Software dazu aufgefordert, geben Sie die Nummer der Registrierung an.

Wurde die Installation erfolgreich abgeschlossen, wird Ihnen dies in einer Meldung auf dem Bildschirm angezeigt.

Falls schon eine Version von SIMATIC Computing installiert ist

Wenn das Installationsprogramm feststellt, daß sich bereits eine WinAC-Installation auf dem Erstellsystem befindet, wird eine entsprechende Meldung angezeigt. Sie haben dann folgende Möglichkeiten fortzufahren:

- Installation abbrechen, um danach die alte Version von WinAC unter Windows NT zu deinstallieren und anschließend die Installation erneut zu starten.
- Installation fortsetzen und damit die alte Version mit der neuen Version überschreiben.

Sie sollten vor einer Installation auf jeden Fall eine eventuell vorhandene ältere Version deinstallieren. Das einfache Überschreiben einer älteren Version hat den Nachteil, daß bei einem anschließenden Deinstallieren eventuell noch vorhandene Teile aus einer älteren Installation nicht gelöscht werden. Wenn Sie die ältere Version von SIMATIC Computing deinstallieren, müssen Sie Ihren Computer neu starten, bevor Sie die neue Version installieren.

Fehler während der Installation

Folgende Fehler führen zum Abbruch der Installation:

- Initialisierungsfehler unmittelbar nach dem Start des Setup-Programms: Sie haben das Programm Setup.exe wahrscheinlich nicht unter Windows NT gestartet.
- Nicht genügend Speicherplatz: Sie benötigen mindestens 20 MB freien Speicherplatz auf Ihrer Festplatte.
- Defekte Diskette: Wenn Sie feststellen, daß eine Diskette defekt ist, wenden Sie sich bitte an Ihre Siemens-Vertretung.
- Bedienerfehler: Beginnen Sie die Installation erneut und beachten Sie die Anweisungen sorgfältig.

Deinstallieren der Software SIMATIC Computing

Gehen Sie folgendermaßen vor, um die Software SIMATIC Computing auf Ihrem Computer zu deinstallieren:

1. Starten Sie unter Windows NT das Dialogfeld zum Installieren von Software, indem Sie in der Systemsteuerung auf das Symbol "Programme installieren/deinstallieren" doppelklicken.
2. Markieren Sie den Eintrag WinAC in der angezeigten Liste der installierten Software. Klicken Sie auf die Schaltfläche zum Deinstallieren der Software.

3.4 Verbinden von SIMATIC Computing mit einer Slot-PLC oder einer Kommunikationskarte

Zum Verbinden von SIMATIC Computing mit einer Slot-PLC oder Kommunikationskarte müssen Sie die Netzverbindung, über die SIMATIC Computing und die Slot-PLC oder die Karte kommunizieren, mit der Anwendung "PG/PC-Schnittstelle einstellen" einrichten.

Hinweis

Sie können nur eine Slot-PLC oder eine Kommunikationskarte zur Zeit anzeigen.

Gehen Sie folgendermaßen vor, um SIMATIC Computing für die Kommunikation mit einer Slot-PLC oder einer Kommunikationskarte einzurichten:

1. Wählen Sie im Startmenü in Windows NT den Befehl **Start > SIMATIC > PC Based Control > WinCP-Konfigurator**.
2. Öffnen Sie das Register "Connection" und wählen Sie die Schaltfläche "PG/PC-Schnittstelle einstellen". Daraufhin öffnet sich das Dialogfeld "PG/PC-Schnittstelle einstellen".

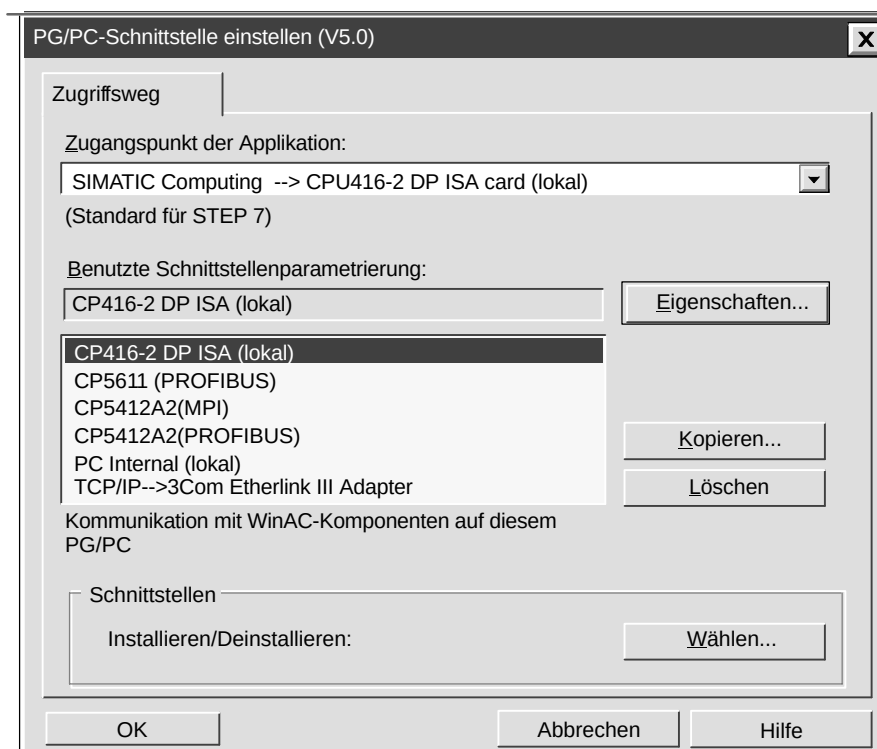


Bild 3-1 Einstellen der PG/PC-Schnittstelle für die Slot-PLC CPU 416-2 DP ISA (lokal)

3. Im aufklappbaren Listenfeld "Zugangspunkt der Applikation" wählen Sie **SIMATIC Computing**.
4. Wählen Sie die Schnittstellenparametrierung, die Ihrem Kommunikationspfad im Netz entspricht, z.B. "CPU416-2 DP ISA (lokal)". Die ausgewählte CPU bzw. Karte wird daraufhin im Feld "Zugangspunkt der Applikation" angezeigt (Bild 3-1).

Zum Einstellen der SIMATIC Computing-Schnittstelle für eine lokale Slot-PLC wählen Sie: **SIMATIC Computing -> <Kartenname> (lokal)**

Zum Einstellen der SIMATIC Computing-Schnittstelle für ein S7-System in einem TCP/IP LAN wählen Sie: **SIMATIC Computing -> <Kartenname> (TCP/IP)**

Zum Einstellen der SIMATIC Computing-Schnittstelle für ein S7-System in einem Industrial Ethernet (ISO-Transportprotokoll) wählen Sie:
SIMATIC Computing -> <Kartenname> (ISO Transport)

Zum Einstellen der SIMATIC Computing-Schnittstelle für ein S7-System in einem PROFIBUS-Netz wählen Sie:
SIMATIC Computing -> <Kartenname> (PROFIBUS).

Zugreifen auf Daten mit SIMATIC Computing

4

Kapitelübersicht

Mit SIMATIC Computing können Sie auf Daten in Control Engines (entweder in WinLC von WinAC Basis, in einer Slot-PLC wie der CPU 416-2 DP ISA von WinAC Pro oder in anderen S7-Automatisierungssystemen) zugreifen. Diese Control Engines können auf dem gleichen Computer wie SIMATIC Computing installiert sein. SIMATIC Computing kann aber auch über ein lokales Netz auf die Control Engine zugreifen.

Hinweis

(Der Begriff "Slot-PLC" in diesem Handbuch bezieht sich auf eine Slot-PLC wie die CPU 416-2 DP ISA oder die CPU 416-2 DP ISA Lite. Im Handbuch fällt die CPU 416-2 DP ISA Lite unter die Beschreibung der CPU 416-2 DP ISA).

Mit SIMATIC Computing können Sie in einem lokalen Netz (LAN) über das Distributed Component Object Model (DCOM) von Windows NT kommunizieren. Über DCOM können Sie dezentrale Anwendungen über ein LAN integrieren. Eine dezentrale Anwendung besteht aus mehreren Prozessen oder unterschiedlichen Computern, die miteinander kooperieren, um eine Aufgabe gemeinsam zu lösen.

SIMATIC Computing kann auch mit einer lokalen oder entfernten Control Engine in einem MPI-, PROFIBUS-DP- oder H1-Netz kommunizieren. Sie können die Verbindung zur Control Engine in der Anwendung "PG/PC-Schnittstelle einstellen" einrichten.

Kapitel	Beschreibung	Seite
4.1	Zugreifen auf Daten in Control Engines	4-2
4.2	Zugreifen auf eine lokale Control Engine	4-4
4.3	Zugreifen auf eine entfernte Control Engine	4-5
4.4	Kommunizieren mit mehreren Control Engines	4-6

4.1 Zugreifen auf Daten in Control Engines

Eine "Control Engine" ist ein Prozessor oder ein Programm, das Daten, die einen Prozeß oder eine Maschine steuern, verwaltet und bearbeitet. Bei der Control Engine kann es sich um Software oder um Hardware handeln. Wie Sie in Bild 4-1 sehen, können Sie über die Elemente der Software SIMATIC Computing auf Daten in folgenden Control Engines zugreifen:

- Windows Logic Controller (WinLC), ein S7-Software-Controller, der im RAM-Speicher Ihres Computers läuft. WinLC kommuniziert mit Ein-/Ausgabebaugruppen über ein PROFIBUS-DP-Netz. Ab Version 3.0 kann SIMATIC Computing über PROFIBUS-DP-, MPI- und H1-Netze auf Daten in WinLC zugreifen.
- Slot-PLC wie die CPU 416-2 DP ISA, ein S7-Controller, der in eine ISA-Karte integriert ist, die in Ihren Computer eingebaut ist. Diese CPU kommuniziert mit Ein-/Ausgabebaugruppen über ein PROFIBUS-Netz.
- Andere S7-CPU's in einem MPI-, H1- oder PROFIBUS-Netz. Diese CPU's verfügen über lokale und dezentrale (entfernte) Ein- und Ausgänge.

SIMATIC Computing verfügt über SIMATIC Controls, die mit Hilfe der Microsoft ActiveX-Technologie Fremd-Anwendungen (z.B. Microsoft Excel oder Visual Basic) Zugriff auf Daten in der Control Engine gewähren. Neben diesen SIMATIC Controls besitzt SIMATIC Computing einen Server, der anderen Anwendungen ein OPC-Server Interface (OLE for Process Control) zur Verfügung stellt.

Mit einer Symboldatei können Sie symbolisch auf Daten auf einem PC zugreifen, auf dem STEP 7 nicht installiert ist. Der Symboldatei-Konfigurator erstellt die Symboldateien aus der Symboltabelle in STEP 7. Die Symboldatei ermöglicht es außerdem, Ihre Anwendung mit mehreren Control Engines gleichzeitig zu verbinden (siehe Abschnitt 4.4).

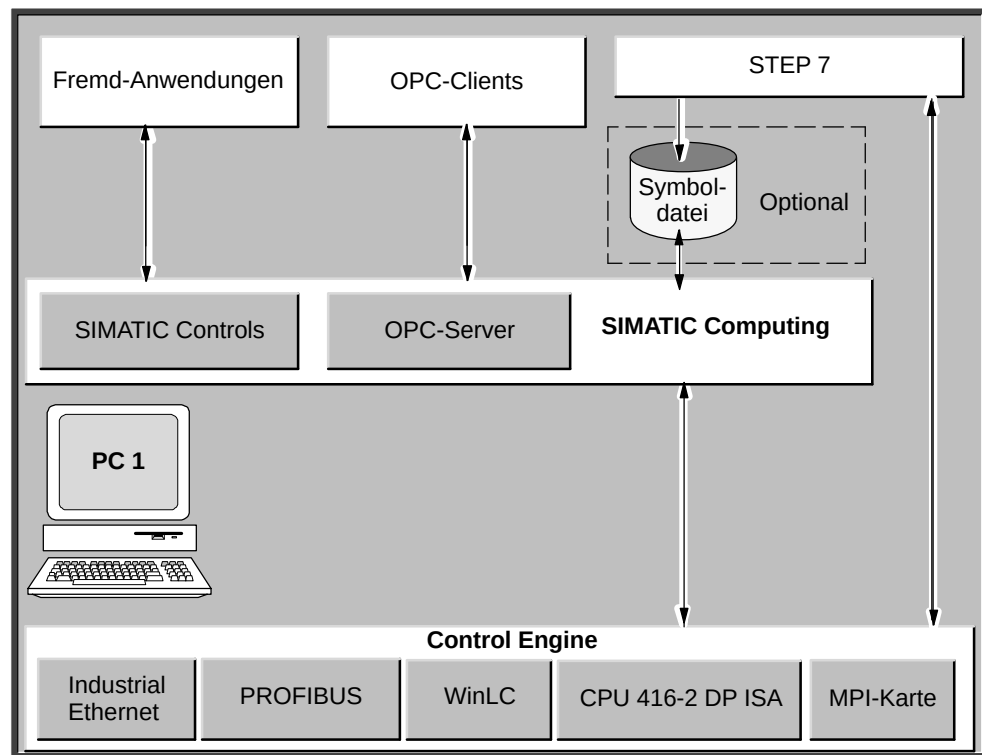


Bild 4-1 Zugreifen auf Daten in Control Engines

4.2 Zugreifen auf eine lokale Control Engine

In der Basiskonfiguration von SIMATIC Computing laufen alle Komponenten auf einem Computer (siehe Bild 4-2). Die gesamte Software läuft auf dem lokalen Computer und der Zugriff der Anwendungen untereinander ist vereinfacht.

Wenn Sie mit Symboldateien arbeiten, können Sie auf die Daten in der Control Engine über Symbole anstelle von absoluten Adressen zugreifen. Wie Sie in Bild 4-2 sehen, kann Ihr Programm über das Symbol "Zuflußventil" auf E0.0 und über das Symbol "Abflußventil" auf E0.1 in der Control Engine zugreifen. Sie erstellen die Symboldatei mit Hilfe der Symboltabelle in STEP 7 – mit der gleichen Symboltabelle, die Sie erstellt haben, als Sie das Programm für die Control Engine entworfen haben.

Die Symboldatei ermöglicht es außerdem, Ihre Anwendung mit mehreren Control Engines gleichzeitig zu verbinden. Informationen zum Anschließen von mehreren Control Engines entnehmen Sie dem Abschnitt 4.4.

Ausführliche Informationen zum Erstellen von Symboldateien entnehmen Sie dem Kapitel 9.

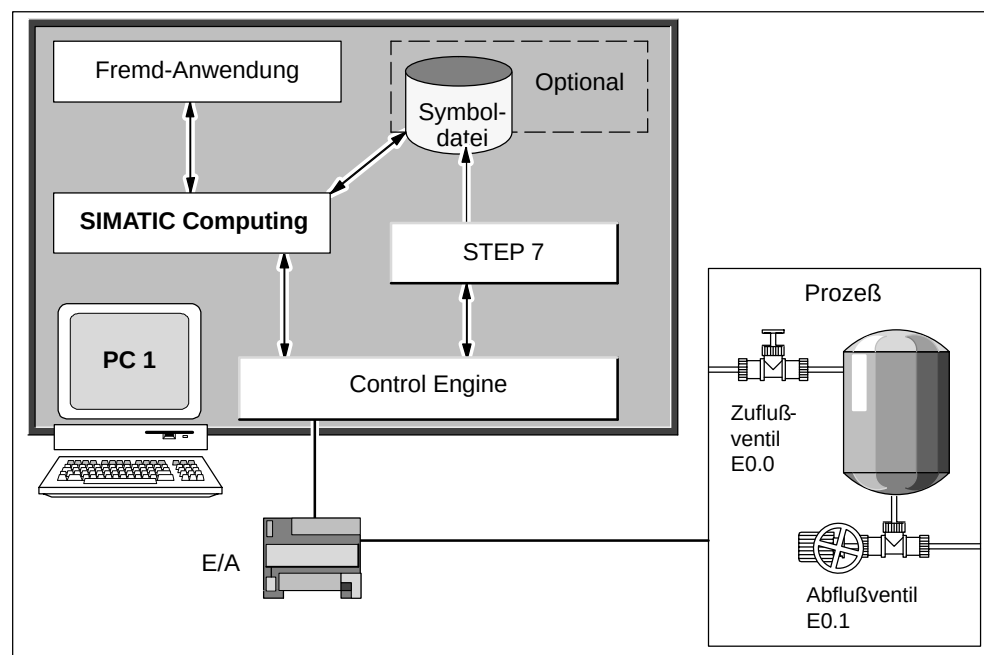


Bild 4-2 Zugreifen auf eine lokale Control Engine

4.3 Zugreifen auf eine entfernte Control Engine

Sie können mit SIMATIC Computing auch auf Control Engines auf entfernten Computern, die sich im lokalen Netz (LAN) befinden, zugreifen. Wie Sie in Bild 4-3 sehen, kann eine Fremd-Anwendung oder eine anwenderspezifische Applikation, die sich auf PC 2 befindet, auf Daten in der Control Engine, die auf PC 1 läuft, zugreifen.

Wenn Sie mit Symboldateien arbeiten, können Sie auf die Daten in der Control Engine über Symbole anstelle von absoluten Adressen zugreifen. Wie Sie in Bild 4-2 sehen, kann Ihr Programm über das Symbol "Zuflußventil" auf E0.0 und über das Symbol "Abflußventil" auf E0.1 in der Control Engine zugreifen. Sie erstellen die Symboldatei mit Hilfe der Symboltabelle in STEP 7 – mit der gleichen Symboltabelle, die Sie erstellt haben, als Sie das Programm für die Control Engine entworfen haben.

Damit Sie symbolische Adressen mit der Anwendung, die auf dem entfernten Computer läuft, verwenden können, müssen Sie die Symboldatei auf den entfernten Computer kopieren. Der entfernte Computer benötigt weder STEP 7 noch die Symboltabelle, um die Symboldatei zu verwenden.

Ausführliche Informationen zum Erstellen von Symboldateien entnehmen Sie dem Kapitel 9.

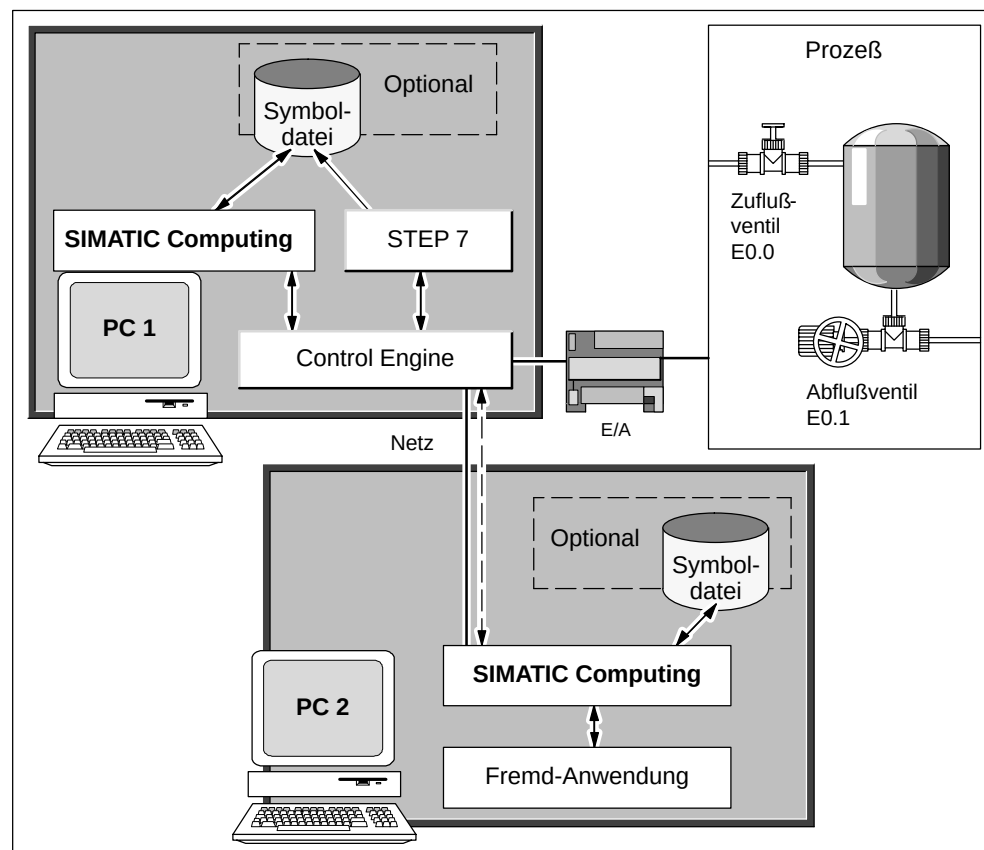


Bild 4-3 Zugreifen auf eine entfernte Control Engine

4.4 Kommunizieren mit mehreren Control Engines

Sie können mit SIMATIC Computing gleichzeitig auf mehrere Control Engines im lokalen Netz zugreifen. Wenn Sie eine Symboldatei erstellen, können Sie das Data Control so einrichten, daß es auf Daten in mehreren Control Engines gleichzeitig zugreift.

Hinweis

Sie können nur eine Slot-PLC oder eine Kommunikationskarte zur Zeit anzeigen.

Bei Einsatz der DCOM-Technologie von Microsoft oder anderen Siemens-Netzen können Sie mit einer Symboldatei auf eine Vielzahl von Control Engines und Anwendungen in Ihrem lokalen Netz zugreifen. Wie Sie in Bild 4-4 sehen, kann ein Computer, auf dem SIMATIC Computing installiert ist (PC 1), mit Hilfe einer Symboldatei auf Daten in den Control Engines, die auf einem anderen Computer laufen (PC 2) sowie auf eine Fremd-Anwendung (z.B. Microsoft Excel oder Visual Basic) zugreifen.

Wie bei einer einzelnen Control Engine (siehe Abschnitte 4.2 und 4.3) bietet die Symboldatei die Möglichkeit, mit symbolischen Adressen auf Daten in mehreren Control Engines zuzugreifen.

Im Anhang E erhalten Sie nähere Informationen dazu, wie Sie SIMATIC Computing über DCOM mit mehreren Control Engines verbinden. Ausführliche Informationen zu Symboldateien und dem Symboldatei-Konfigurator entnehmen Sie dem Kapitel 9.

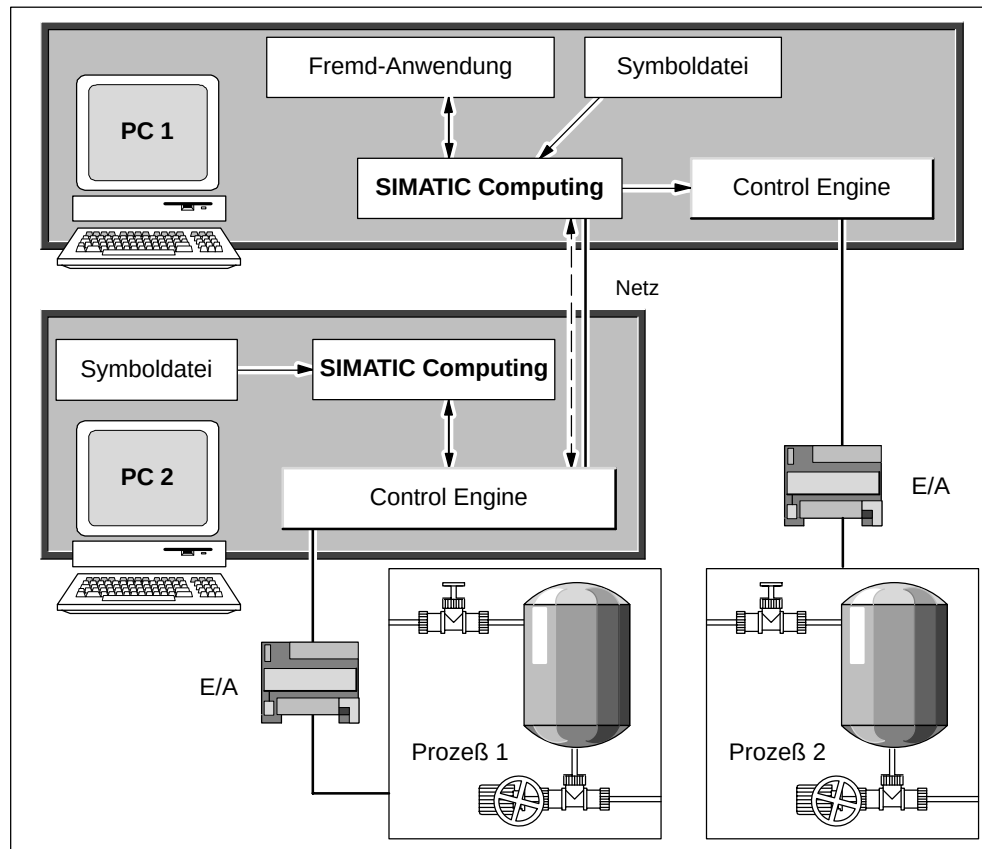


Bild 4-4 Zugreifen auf Daten in mehreren entfernten Control Engines

Zugreifen auf Prozeßdaten mit dem Data Control

5

Kapitelübersicht

Das Data Control verbindet Ihre ActiveX Controls mit der Control Engine (z.B. WinLC oder Slot-PLC wie CPU 416-2 DP ISA).

Das Data Control verfügt über bestimmte Eigenschaften, die Sie einstellen können:

- Im WinAC SoftContainer von SIMATIC Computing: Durch Doppelklick auf das Data Control rufen Sie die Eigenschaften des Controls im Dialogfeld "Eigenschaften" auf. Dieses Dialogfeld enthält die folgenden Register: General, Engine, Connections, Events und Name.
- In anderen Container-Anwendungen (z.B. Microsoft Visual Basic): Sie greifen auf die Eigenschaften so zu, wie auch für andere Controls in dem Container (z.B. über die rechte Maustaste). Öffnen Sie das Kontextmenü des Data Control mit der rechten Maustaste und wählen Sie den Befehl **Eigenschaften**.

Kapitel	Beschreibung	Seite
5.1	Verbinden der SIMATIC Controls mit der Control Engine	5-2
5.2	Einrichten der Verbindungseigenschaften für das Data Control	5-3
5.3	Einstellen der Control Engine für das Data Control	5-4
5.4	Verbinden der ActiveX Controls mit der Control Engine	5-9
5.5	Filtern der Eigenschaften für die ActiveX Controls	5-13
5.6	Einrichten von anwenderspezifischen Ereignissen	5-15
5.7	Erstellen einer Verbindungstabelle	5-16
5.8	Beispielprogramm zum Erstellen einer Verbindungstabelle und einer Ereignistabelle	5-17
5.9	Beispielprogramm zum Reagieren auf Ereignisse	5-19
5.10	Beispielprogramm zum Lesen und Schreiben von Daten	5-24
5.11	Beispielprogramm zum Lesen und Schreiben von booleschen Daten	5-29
5.12	Eigenschaften, Methoden und Ereignisse des Data Control	5-30

5.1 Verbinden der SIMATIC Controls mit der Control Engine



Vorsicht

Wenn Sie die Zeiten in Ihrem Programm nicht deaktivieren, kann es sein, daß zeit-gesteuerte Verbindungen bestehen bleiben, die weiterhin Daten in die Control Engine schreiben. Dadurch ist unvorhersehbarer Betrieb der Control Engine mög-lich, der zu Sachschaden oder Körperverletzung führen kann.

Damit sichergestellt ist, daß alle Verbindungen getrennt sind, wenn Ihr Programm geschlossen wird, müssen Sie im Unterprogramm Form_Unload alle Zeiten vor dem Ausdruck End deaktivieren.

Wenn Sie auf Prozeßdaten zugreifen möchten, müssen die WinAC Controls (Button, Edit, Label und Slider) zunächst eine Verbindung über das Data Control aufbauen. Bild 5-1 zeigt die Beziehung zwischen dem Data Control und den ande-ren SIMATIC Controls.

Hinweis

Das Panel Control (verfügbar mit WinLC oder einer Slot-PLC) und das DBuffer Control (zum Zugreifen auf den Diagnosepuffer von S7-Automatisierungssystemen) stellen die Verbindung zur Control Engine nicht über das Data Control her.

Im Register "Connections" des Data Control ordnen Sie der Eigenschaft Value für jedes Control eine Variable (die Adresse im Speicher) zu. Das Data Control konfi-guriert die Control Engine so, daß die Adressen der zugeordneten Variablen im Speicher in bestimmten Abständen (in ms) geprüft werden. Ändert sich ein Wert, wird der neue Wert in das Data Control geschrieben. Das Data Control schreibt diesen neuen Wert dann in die anderen Controls.

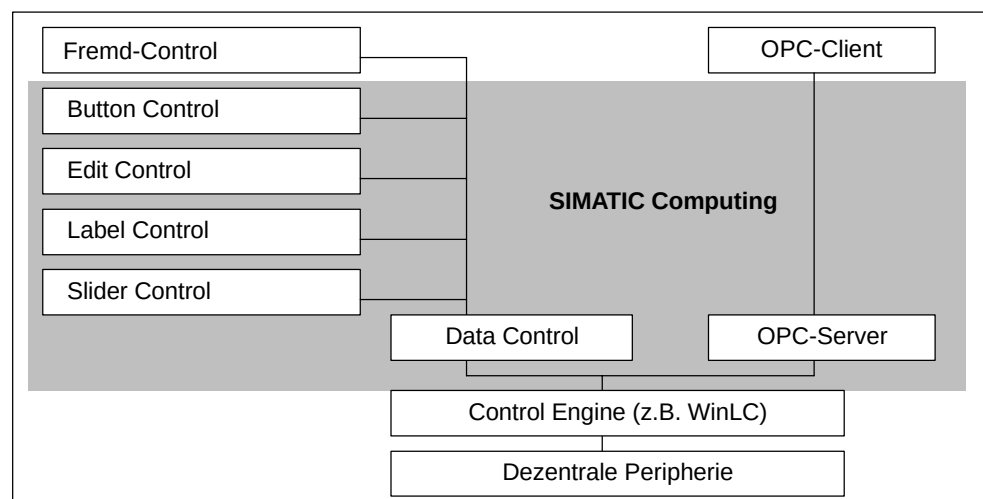


Bild 5-1 Aufbau einer Verbindung zur Control Engine mit dem Data Control

5.2 Einrichten der Verbindungseigenschaften für das Data Control

Im Register "General" können Sie die folgenden Parameter für die Verbindung zur Control Engine konfigurieren (siehe Bild 5-2):

- **AutoConnect (Automatische Verbindung):** Wenn Sie dieses Kontrollkästchen aktivieren, stellt das Data Control automatisch die Verbindung zu den Adressen in der Control Engine her. Ist das Kontrollkästchen deaktiviert, stellt das Data Control nur dann eine Verbindung zu den Adressen im Speicher her, wenn es vom Programmcode (Methode Connect), den Sie dem Control zugeordnet haben, dazu aufgefordert wird.
- **AutoConnect Timeout:** AutoConnect Timeout (ms): Hier geben Sie an, wie lange das Data Control zwischen dem Verbinden mit der Control Engine und dem Schreiben der Daten warten soll.

Es kann sein, daß in einigen Containern kein Mechanismus vorhanden ist, der dem Data Control mitteilt, daß es in die Control Engine schreiben soll. Das Data Control beginnt nach Ablauf der von Ihnen für Timeout angegebenen Zeit, Daten zu schreiben.

- **Default Update Rate (ms):** Hier geben Sie an, in welchen Abständen (in ms) die Control Engine die Adressen im Speicher auf Änderungen prüft.
- **Default Dead Band:** Hier geben Sie der Control Engine das Ausmaß der Änderungen an, die in einem Wert auftreten müssen, damit die Control Engine den neuen Wert in das Data Control schreibt. Beispiel: Haben Sie für die Totzone einen Wert von 10 angegeben und der Wert in der Control Engine beträgt 22, dann schreibt die Control Engine erst dann einen neuen Wert, wenn der Wert 33 oder 11 beträgt.
- **Show Error Boxes:** Hier geben Sie an, ob bei Fehlern, die vom Anwender hervorgerufen werden, die Standard-Fehlermeldungen angezeigt werden sollen. SIMATIC Computing verfügt nur über Fehlermeldungen in englischer Sprache. Wenn Sie Meldungen in anderen Sprachen anzeigen möchten, müssen Sie diese Option deaktivieren und einen Programmcode für die Reaktion auf das Fehlerereignis schreiben.

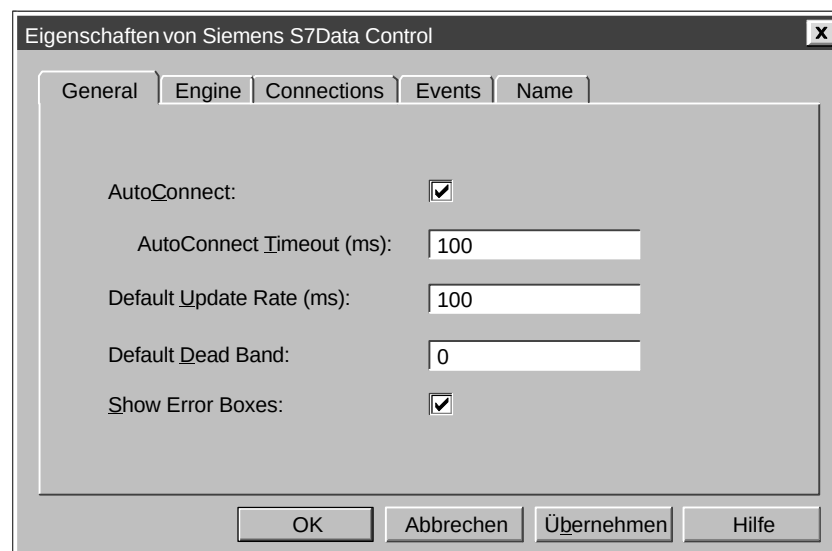


Bild 5-2 Eigenschaften des Data Control (Register "General")

5.3 Einstellen der Control Engine für das Data Control

In SIMATIC Computing können Sie eine Verbindung zu einer oder zu mehreren Control Engines herstellen. Sie können die Verbindung zur Control Engine auch über ein Netz herstellen, z.B. ein lokales Datennetz (LAN). Im Register "Engine" (Bild 5-3) im Dialogfeld "Eigenschaften" stellen Sie die Control Engine ein.

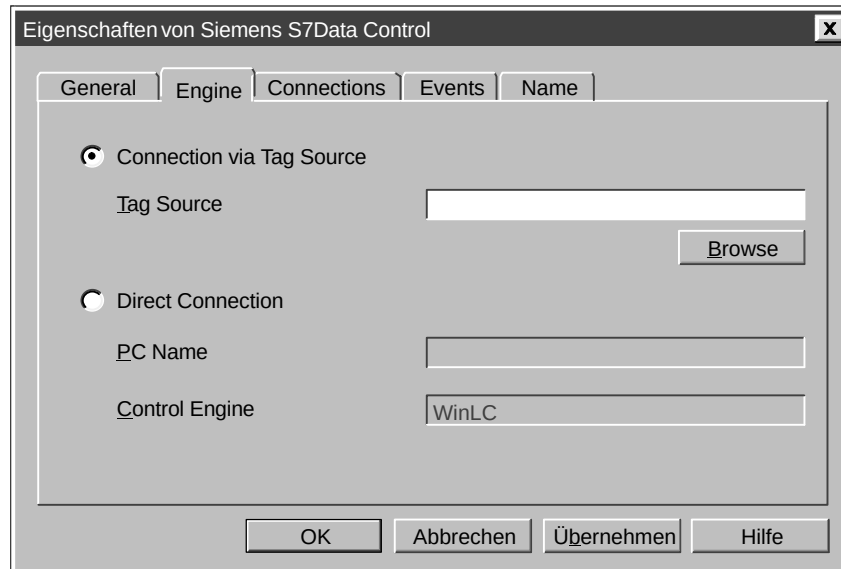


Bild 5-3 Eigenschaften des Data Control (Register "Engine")

Bei einer Control Engine kann es sich um eine Slot-PLC wie die CPU 416-2 DP ISA, eine S7-CPU oder um WinLC handeln. Es gibt zwei Arten von Zeichenfolgen der Control Engine für das S7 Data Control:

- Direkt an WinLC
- Mittels Siemens-Netz

Die Zeichenfolge der Control Engine für eine direkte Verbindung mit WinLC lautet **WinLC**. Dies ist eine COM-Verbindung. Sie brauchen sie nicht in der Anwendung "PG/PC-Schnittstelle einstellen" zu konfigurieren.

Verbindungen zu den verschiedenen Arten von Netzen werden mit der Anwendung "PG/PC-Schnittstelle einstellen" folgendermaßen konfiguriert:

- Zugriff auf das MPI-Netz: SIMATIC Computing--> <Kartenname> (MPI)
- Zugriff auf das DP-Netz: SIMATIC Computing--> <Kartenname> (PROFIBUS)
- Zugriff auf das H1-Netz: SIMATIC Computing--> TCP/IP--> <Kartenname> (Für den Zugriff auf das Netz müssen das NCM-Optionspaket und STEP 7 V5 SP3 installiert werden.)

Hinweis

Damit Sie WinLC als Control Engine einsetzen können, müssen Sie auch die Verbindung in WinLC einrichten. Informationen hierzu entnehmen Sie dem Kapitel über Verbindungen von STEP 7 mit WinLC oder Automatisierungssystemen im Handbuch *SIMATIC Windows Logic Controller (WinLC)*.

Einsetzen einer Symboldatei

Die Symboldatei enthält symbolische Namen, die das Data Control einsetzen kann, um auf Daten oder Control Engines zuzugreifen. Der Einsatz einer Symboldatei bietet folgende Möglichkeiten:

- Sie können das Data Control mit verschiedenen Control Engines gleichzeitig verbinden.
- Sie können mit Hilfe der in STEP 7 definierten Symbole auf Adressen im Speicher der Control Engine zugreifen.

Die Symboldatei dient auch als Hilfestellung beim Eingeben von Symbolen. Ausführliche Informationen zu Symboldateien entnehmen Sie dem Kapitel 9.

Informationen zum Anschließen von mehreren Control Engines entnehmen Sie dem Abschnitt 4.4. In Abschnitt 9.1 erfahren Sie weitere Informationen zum Einsetzen von STEP 7 und dem Symboldatei-Konfigurator für Verbindungen mit entfernten Control Engines.

Gehen Sie folgendermaßen vor, um das Data Control mit einer in einer Symboldatei definierten Control Engine zu verbinden:

1. Doppelklicken Sie auf das Data Control (oder wählen Sie den Menübefehl **Bearbeiten > Eigenschaften**), um das Dialogfeld "Eigenschaften" des Data Control aufzurufen.
2. Öffnen Sie das Register "Engine", um die Konfigurationsmöglichkeiten anzuzeigen.
3. Wählen Sie die Option "Connection via Tag Source".
4. Wählen Sie die Schaltfläche "Browse" und markieren Sie die gültige Symboldatei (*.tsd) (siehe Bild 5-4).
5. Klicken Sie auf die Schaltfläche "Übernehmen", um das Data Control für die Verbindung zur Control Engine über die entsprechende Symboldatei einzurichten.

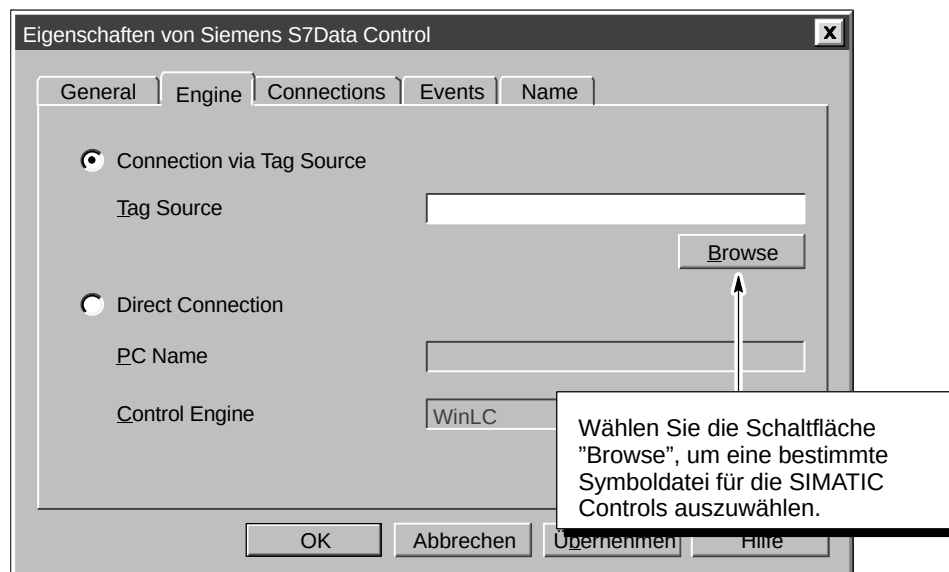


Bild 5-4 Einstellen einer Symboldatei für das Data Control

Herstellen einer Verbindung zu einer bestimmten Control Engine

Wie Sie in Bild 5-5 sehen, können Sie Ihr Programm mit dem Data Control mit einer Control Engine verbinden, die sich auf dem lokalen Computer oder auf einem anderen Computer befindet.

Wenn Sie die Verbindung zu einer Control Engine über ein LAN über DCOM herstellen, müssen Sie im Feld "PC Name" auch den Netznamen des Server-PC angeben. Wenn Sie die Option "Direct Connection" wählen, müssen Sie im Feld "Control Engine" den Namen der Control Engine angeben. Die Zeichenfolgen der Control Engine unterstützen die Adresse von Baugruppenträger und Steckplatz der CPU am Netzknoten und umfassen H1-Netze und IP- und MAC-Adressen. Tragen Sie folgende Zeichenfolgen im Feld "Control Engine" ein:

- **<lokal>** (So geben Sie die Control Engine an, die sich auf dem gleichen Computer wie SIMATIC Computing befindet.)
- **WinLC** (So geben Sie WinLC von WinAC Basis an.)
- **wsS7=3** (So geben Sie eine Slot-PLC wie die CPU 416-2 DP ISA von WinAC Pro an.)
- **wsS7=xx, a, b** (So geben Sie andere Automatisierungssysteme im MPI-Netz an, Erläuterung: xx ist die MPI-Adresse, a ist die Nummer des Baugruppenträgers und b ist die Nummer des Steckplatzes.) WinLC hat immer folgende Zuweisungen: Baugruppenträger 0, Steckplatz 2.
- **wcIP=xxx.xxx.xxx.xxx,a,b** bei einer Control Engine in einem TCP/IP LAN oder **wcMAC-xx.xx.xx.xx.xx,a,b** bei einer Control Engine in einem Industrial Ethernet mit STEP 7 V5 SP3. Ausführliche Informationen zu den Zeichenfolgen der Control Engine entnehmen Sie dem Anhang G.

Einige Zeichenfolgen wurden vereinfacht, damit Sie besser eingegeben werden können, doch das Data Control nimmt auch längere Zeichenfolgen von Anwendungen an, die mit früheren Versionen von SIMATIC Computing erstellt wurden (z.B. S7DosIntf\MPI=3).

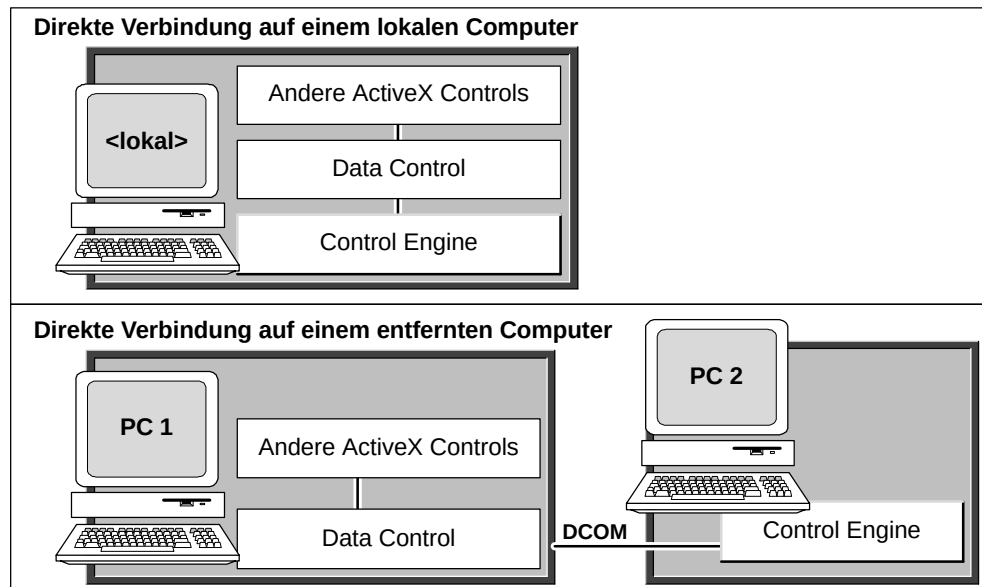


Bild 5-5 Direkte Verbindung für einen lokalen oder einen entfernten Computer

Hinweis

Wenn Sie das Data Control für eine Verbindung mit einer (bestimmten) Control Engine einrichten, können Sie keine Symboldatei zuordnen. D.h. Sie können keine symbolischen Namen für die Variablen in der Control Engine verwenden.

Um mit symbolischen Namen zu arbeiten, wählen Sie die Option zum Verbinden mehrerer Control Engines und geben eine Symboldatei an, die Symbole für eine Control Engine enthält.

Gehen Sie folgendermaßen vor, um das Data Control für eine Verbindung mit einer bestimmten Control Engine einzurichten:

1. Doppelklicken Sie auf das Data Control (oder wählen Sie den Menübefehl **Bearbeiten > Eigenschaften**), um das Dialogfeld "Eigenschaften" des Data Control aufzurufen.
2. Öffnen Sie das Register "Engine", um die Konfigurationsmöglichkeiten anzuzeigen.
3. Wählen Sie die Option "Direct Connection" (siehe Bild 5-6).
4. Wenn Sie die Verbindung zu einer Control Engine auf dem lokalen Computer herstellen möchten, tragen Sie in das Feld "PC Name" die Zeichenfolge **<lokal>** ein. Sie können nicht gleichzeitig auf WinLC und eine Slot-PLC zugreifen.

Zum Verbinden einer Control Engine auf einem entfernten Computer:

- Tragen Sie im Feld "PC Name" den Netznamen des Server-PCs ein (z.B. "PC_2").
- Tragen Sie im Feld "Control Engine" den Namen der Control Engine, z.B. **WinLC** oder **ws7=3** (für eine Slot-PLC wie die CPU 416-2 DP ISA von WinAC Pro) ein.

5. Wählen Sie die Schaltfläche "Übernehmen", um das Data Control einzurichten.

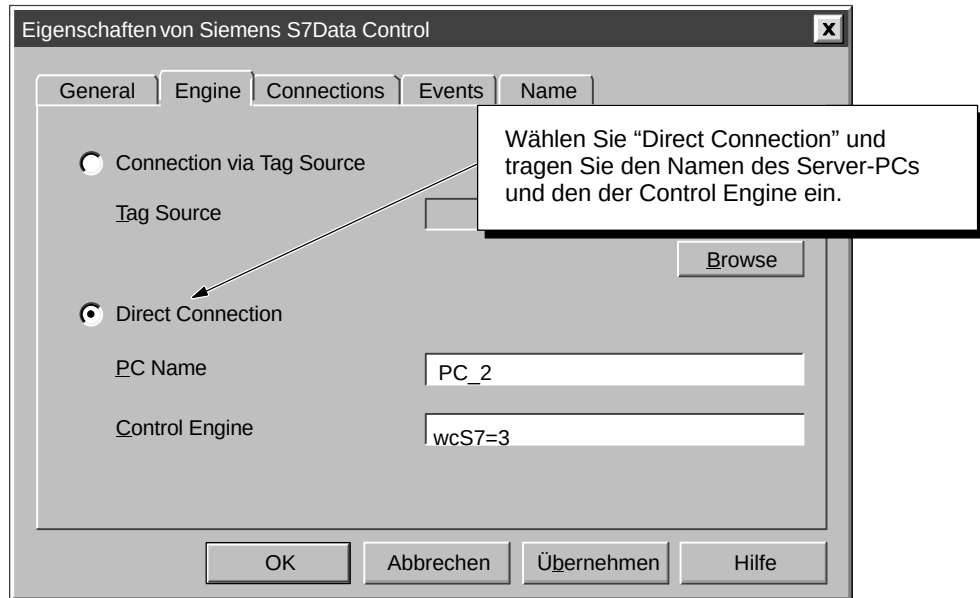


Bild 5-6 Einrichten von DCOM für eine bestimmte Control Engine

Hinweis

Damit Sie über LAN eine Verbindung zu einem entfernten Computer herstellen können, müssen Sie die jeweiligen Computer für DCOM eingerichtet haben. In den Abschnitten E.2 und E.3 erfahren Sie, wie Sie die Server- und Client-Computer für DCOM einrichten.

5.4 Verbinden der ActiveX Controls mit der Control Engine

Das Register "Connections" zeigt die ActiveX Controls (und ob es sich um SIMATIC Controls oder um Fremd-Controls handelt), die mit der Control Engine verbunden werden können.



Vorsicht

Wenn Sie die Zeitfunktion oder Haltepunkte in Visual Basic mit SIMATIC Computing verwenden, kann dies zu Fehlern führen, die evt. Ihren Computer bzw. die Anwendung zum Absturz bringen bzw. verriegeln können. Je nach Konfiguration kann die Anwendung dadurch die Kommunikation mit der Control Engine beenden. Dies kann unvorhersehbare Reaktionen im Prozeß auslösen, die zu Tod, schwerer Körperverletzung und/oder Sachschaden führen können.

Installieren Sie einen physikalischen NOT-AUS-Schaltkreis für die Maschine bzw. den Prozeß.

Timer in VB: Bei der Timer-Funktion in Visual Basic Version 5 kann ein Timer den in der gleichen Einweg-Bearbeitung befindlichen Code unterbrechen, wodurch Fehler mit schwerwiegenden Folgen auftreten können. Beachten Sie folgende Richtlinien, wenn Sie VB-Timer mit SIMATIC Computing einsetzen:

- Deaktivieren Sie die Timer immer im Unterprogramm Form_Unload. Andernfalls kann ein Timer ein Ereignis auslösen, während das VB-Programm geschlossen wird. Diese Bedingung kann einen Absturz bzw. eine Verriegelung Ihres Computers oder der Anwendung herbeiführen. Es kann auch sein, daß die Anwendung unsichtbar weiterläuft.
 - Wenn Sie den Timer im Unterprogramm Form_Load starten, kann es sein, daß das Zeitereignis auftritt, bevor die anderen Objekte instanziiert wurden. Um sicherzustellen, daß die Objekte korrekt instanziiert werden, starten Sie einen Timer im Unterprogramm Form_Load immer mit einem großen Intervall (z.B. 1 oder 2 Sekunden). Nachfolgende Timer-Intervalle können kürzer sein.
-

Zuordnen einer Variablen zu einer Eigenschaft eines Control

Zum Definieren einer Verbindung in der Control Engine ordnen Sie einer Eigenschaft eines Control eine Variable (eine Adresse im Speicher) der Control Engine zu (siehe Bild 5-7).

Hinweis

In SIMATIC Computing können Sie nicht in Zeiten in der Control Engine schreiben.

Gehen Sie folgendermaßen vor, um einer Eigenschaft des Controls eine Variable in der Control Engine zuzuordnen:

1. Im Register "Connections" des Data Control wählen Sie den Namen der Eigenschaft.
2. Tragen Sie in das Feld "Assigned Variable" die Adresse im Speicher der Control Engine ein.

Hinweis

Haben Sie dem Data Control eine Symboldatei zugeordnet, können Sie Symbole anstelle von absoluten Adressen eintragen. Sie können auch über die Schaltfläche "Browse" das Symbol suchen. Wenn Sie einen Punkt "." eingeben, wird eine Liste der gültigen Symboldateien und Symbole jeder Hierarchie angezeigt.

- Haben Sie für die Control Engine die Option "Direct Connection" eingestellt (keine Verwendung von Symbolen), geben Sie die absolute Adresse (z.B. **MB0**) für die Adresse im Speicher der Control Engine an. Im Anhang A erhalten Sie Informationen zu den Datentypen und Speicherbereichen von S7-Automatisierungssystemen.
 - Haben Sie die Option "Connection via Tag Source" für den Einsatz von mehreren Control Engines eingestellt (doch Sie arbeiten **nicht** mit Symbolen), tragen Sie die absolute Adresse (z.B. **MB0**) für die Adresse im Speicher der voreingestellten Control Engine ein. Im Anhang A erhalten Sie Informationen zu den Datentypen und Speicherbereichen von S7-Automatisierungssystemen. Sie können auch eine absolute variable Adresse an den symbolischen Namen einer Control Engine anhängen (z.B. ce1.mb0, ce2.mb0). Informationen zum Einstellen der voreingestellten Control Engine entnehmen Sie dem Abschnitt zum Verwenden von absoluten Adressen mit Symboldateien in Abschnitt 9.3.
 - Haben Sie die Option "Connection via Tag Source" zur Verwendung von Symbolen eingestellt (ganz gleich, ob Sie nur eine oder mehrere Control Engines verbinden), tragen Sie die symbolische Adresse ein (z.B. **Start_Prozess**). Sie können die Adresse an den symbolischen Namen der Control Engine anhängen (z.B. **PC_2_WinLC**).
3. Wählen Sie die Schaltfläche "Übernehmen", um die Variable der Eigenschaft zuzuordnen.

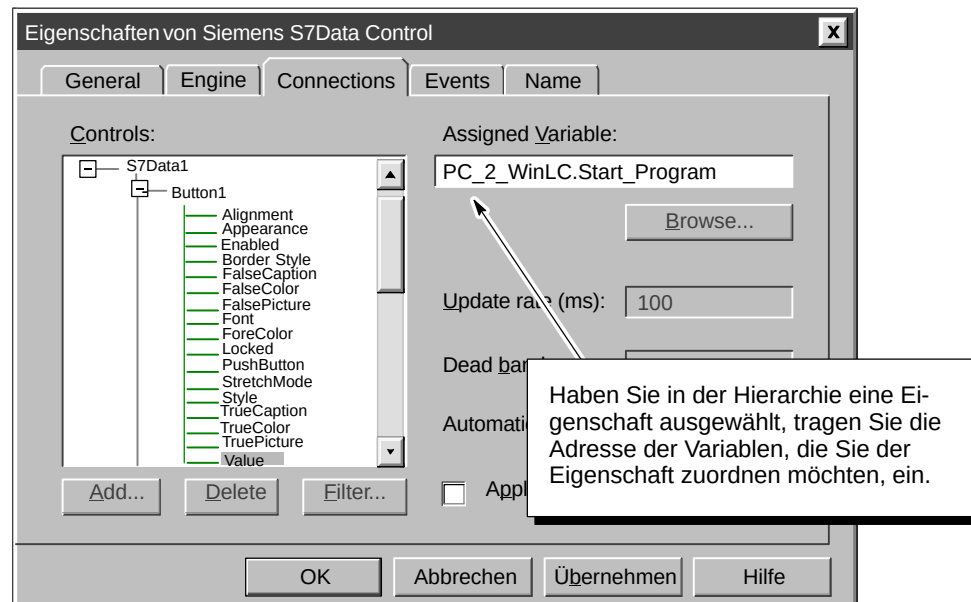


Bild 5-7 Eingeben eines Symbols für die zugeordnete Variable

Wenn Sie eine Symboldatei definiert haben, können Sie nach einem Symbol für die Variable suchen. Klicken Sie auf die Schaltfläche "Browse" (siehe Bild 5-8) und wählen Sie die Variable aus den Symbolen aus, die für die Symboldatei aufgeführt werden. Wählen Sie ein Symbol und drücken Sie die Eingabetaste. (Bild 5-8 zeigt eine gefilterte Liste der Eigenschaften, damit nur die Eigenschaften Enabled und Value angezeigt werden. Ausführliche Informationen zum Filtern der Eigenschaftsliste entnehmen Sie dem Abschnitt 5.5.)

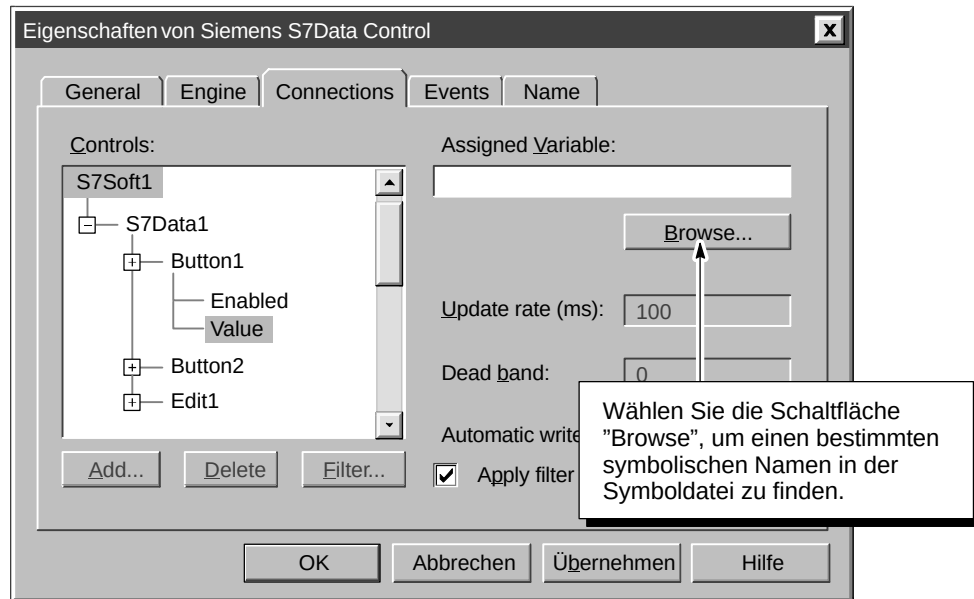


Bild 5-8 Suchen eines Symbols in der Symboldatei

Eingeben von absoluten Adressen anstelle von Symbolen

Wenn Sie dem Data Control eine Symboldatei zugewiesen haben, können Sie anstelle von symbolischen Namen, die in der Symboltabelle in STEP 7 erstellt wurden, auch absolute Adressen eingeben. Suchen Sie das Symbol über die Schaltfläche "Browse". Wenn Sie im Feld "Assigned Variable" ein Zeichen eingeben, wird eine Liste gültiger Symbole der ersten Ebene angezeigt (Eingabehilfe). Wenn Sie eines dieser Symbole auswählen und dann einen Punkt "." eingeben, werden die Symbole der nächsten Ebene angezeigt. Gehen Sie folgendermaßen vor, um absolute Adressen anstelle von symbolischen Namen einzugeben:

1. Tragen Sie das erste Zeichen der absoluten Adresse ein.
2. Drücken Sie die Taste Esc (Escape).
3. Tragen Sie den Rest der absoluten Adresse ein.

Ergänzen einer Verbindung

Wenn Sie für ein ActiveX Control eine Verbindung konfigurieren möchten, bevor Sie das Control in Ihrem ActiveX Container ablegen, können Sie mit der Schaltfläche "Add" eine Instanz des Controls in die Liste der Controls aufnehmen. Klicken Sie auf die Schaltfläche "Add", um die Instanz anzugeben, die Sie mit dem Data Control verbinden möchten.

Nachdem Sie die Instanz des ActiveX Controls in die Liste der Controls aufgenommen haben, können Sie die Instanz in der Liste markieren, die Schaltfläche "Add" erneut anwählen und zusätzliche Eigenschaften ergänzen. Sie können beispielsweise eine Instanz eines Edit Control in die Liste der Verbindungen aufnehmen und anschließend dem Edit Control die Eigenschaft Value hinzufügen, um dem Edit Control eine Variable zuzuordnen zu können.

Löschen einer Verbindung

Wenn Sie im ActiveX Container ein Control löschen, dann bleibt die Verbindung im Data Control konfiguriert. Das bedeutet, daß beim nächsten Mal, wenn Sie ein Control gleichen Namens in den gleichen Container einfügen, die Verbindung, die Sie für das vorherige Control konfiguriert haben, automatisch dem neuen Control zugeordnet wird. Wenn Sie beispielsweise ein Control namens Edit1 löschen und später ein neues Edit Control einfügen, ist der voreingestellte Name für das Control Edit1, und das neue Control übernimmt die bestehende Verbindung für Edit1. Arbeiten Sie mit der Schaltfläche "Delete", wenn Sie nicht möchten, daß neue Controls zuvor konfigurierte Verbindungen übernehmen. Wählen Sie hierzu in der Liste der Controls die Instanz, dessen Verbindung Sie löschen möchten, und klicken Sie auf die Schaltfläche "Delete".

Hinweis

Wenn Sie ein Control löschen oder ein neues Control doch nicht einfügen möchten, auch wenn Sie dafür schon über die Schaltfläche "Add" eine Verbindung konfiguriert haben, können Sie die Verbindung zum Control nur löschen, wenn es im ActiveX Container kein Control gibt, das den in der Verbindung angegebenen Namen verwendet. Löschen Sie die Verbindung, bevor Sie ein neues Control ergänzen, das den Namen, der in der Verbindung angegeben ist, verwendet. Sie können eine Verbindung zu einem Control, das sich im ActiveX Container befindet, nicht über die Schaltfläche "Delete" löschen.

5.5 Filtern der Eigenschaften für die ActiveX Controls

Das Data Control verfügt über einen Filter, mit dem Sie sich eine Untermenge der Eigenschaften des Control anzeigen lassen können. Beispiel: Sie möchten sich nur die Eigenschaften Enabled und Value anzeigen lassen und nicht durch alle anderen Eigenschaften des Control blättern.

Gehen Sie zum Filtern von Eigenschaften folgendermaßen vor:

1. Rufen Sie das Dialogfeld "Eigenschaften" für das Data Control auf.
2. Wählen Sie die Schaltfläche "Filter" (siehe Bild 5-9).

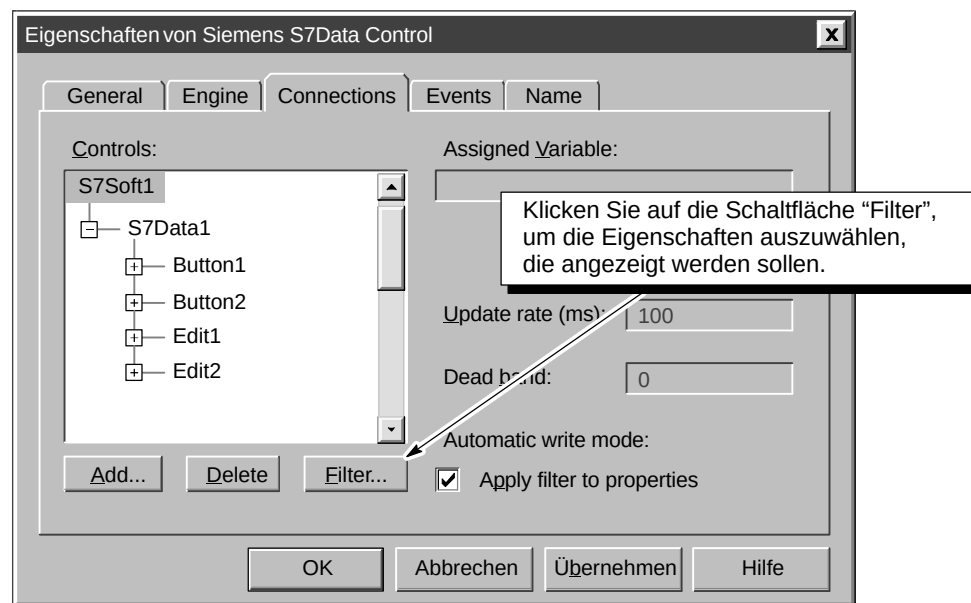


Bild 5-9 Eigenschaften des Data Control (Register "Connections")

3. Geben Sie die Eigenschaften, die angezeigt werden sollen, ein und wählen Sie die Schaltfläche "Add" (siehe Bild 5-10). Mit der Schaltfläche "Edit" können Sie Einträge korrigieren und mit der Schaltfläche "Delete" löschen Sie Einträge.
4. Aktivieren Sie das Kontrollkästchen "Apply filter to properties" (siehe Bild 5-11), um die gefilterten Eigenschaften anzuzeigen.

Mit dem Kontrollkästchen "Apply filter to properties" können Sie den Filter ein- und ausschalten.

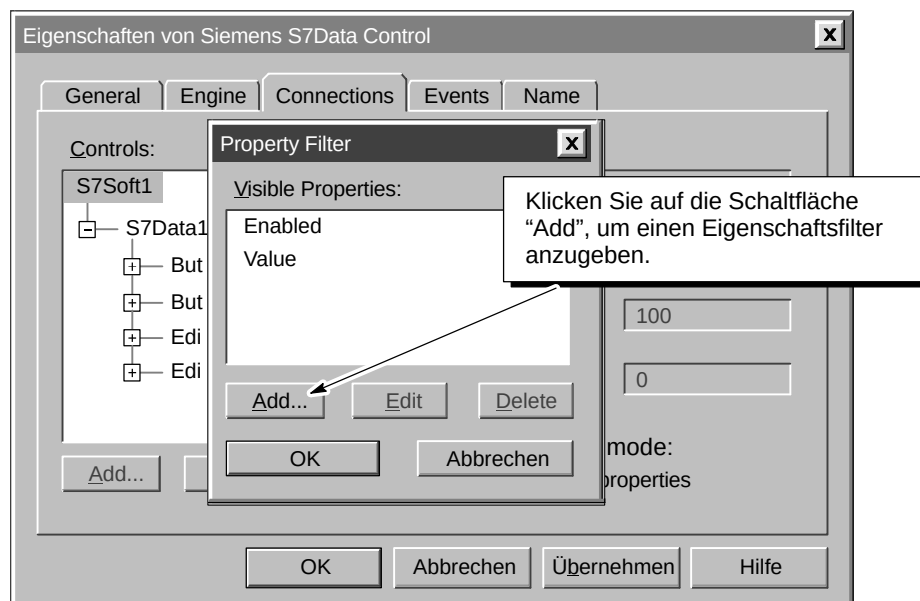


Bild 5-10 Eigenschaften des Data Control (Register "Connections")

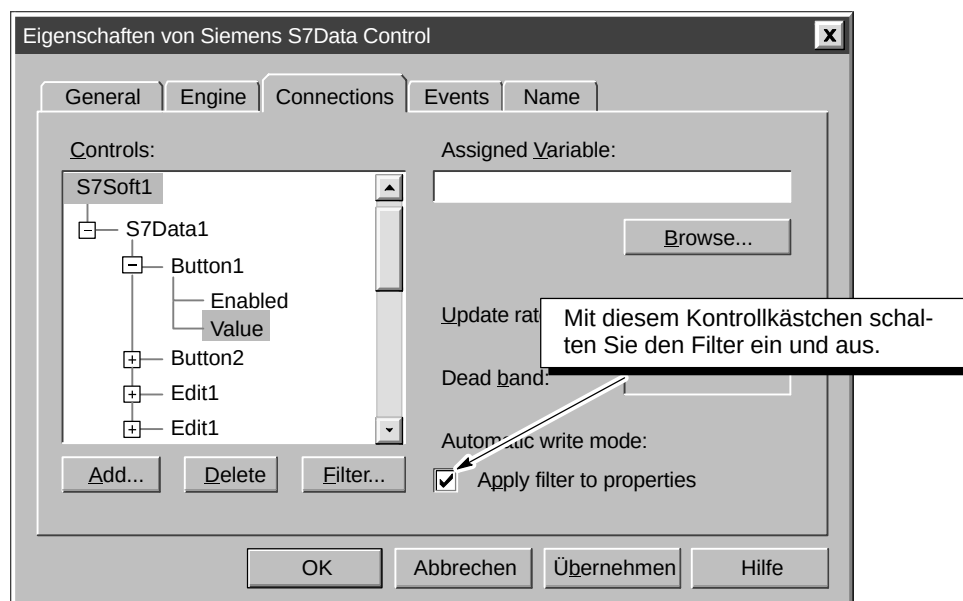


Bild 5-11 Eigenschaften des Data Control (Register "Connections")

5.6 Einrichten von anwenderspezifischen Ereignissen

In Bild 5-12 sehen Sie das Register "Events", in dem Sie anwenderspezifische Ereignisse ergänzen können, die vom Data Control ausgelöst werden. Sie geben eine Zeichenkette (Key) an und ordnen dieser Zeichenkette eine Adresse im Speicher (Variable) zu. Ändert sich diese Variable, erzeugt das Data Control ein Ereignis mit einem Parameter, der die Zeichenkette enthält, die Sie in dem Feld "Key" angegeben haben. Ihr Programm kann dann auf dieses Ereignis reagieren. Im Abschnitt 5.9 finden Sie ein Beispielprogramm, das auf Ereignisse in der CPU reagiert.

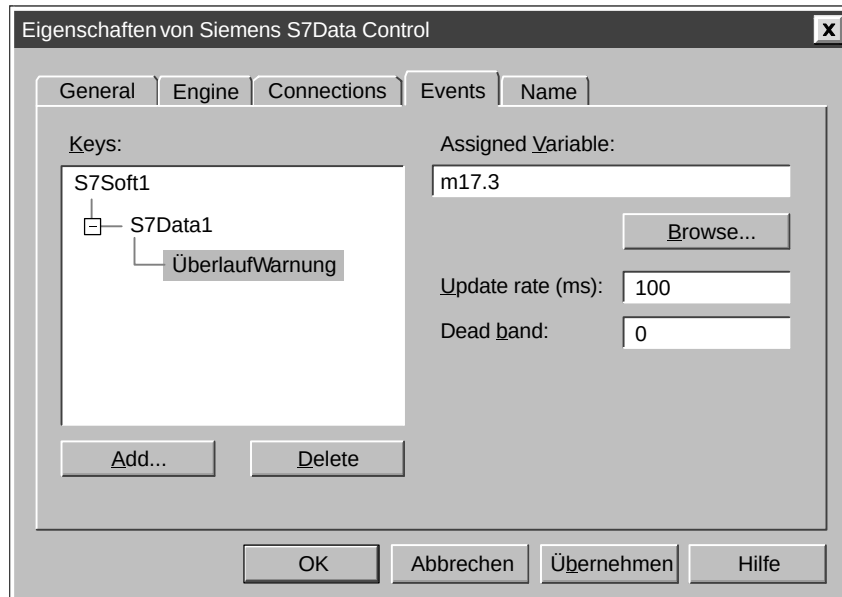


Bild 5-12 Eigenschaften des Data Control (Register "Events")

Ergänzen eines Ereignisses

Mit der Schaltfläche "Add" können Sie anwenderspezifische Ereignisse definieren, die auf Änderungen von Werten im Controller reagieren. Sie können eigenen Code schreiben, der das Ereignis bearbeitet. Hierzu verwenden Sie das Ereignis Value-Changed eines Data Control. Markieren Sie das Data Control in der Liste im Feld "Keys" und wählen Sie die Schaltfläche "Add". Geben Sie dann einen beliebigen Namen für das Ereignis ein (z.B. ÜberlaufWarnung). Anschließend tragen Sie eine Variable in das Feld "Assigned Variable" ein, um den Prozeßwert anzugeben, bei dem das Ereignis ausgelöst werden soll.

Bild 5-12 zeigt ein Beispiel für ein ergänztes Ereignis. Ändert sich der Wert M17.3, wird der Handler für das Ereignis im Data Control aufgerufen. Eingegeben wird in den Handler des Ereignisses die Zeichenkette **ÜberlaufWarnung**.

Löschen eines Ereignisses

Zum Löschen eines anwenderdefinierten Ereignisses öffnen Sie die Liste im Feld "Keys", wählen das gewünschte Ereignis aus und klicken auf die Schaltfläche "Delete".

5.7 Erstellen einer Verbindungstabelle

Das Data Control setzt eine Verbindungstabelle ein, um zu ermitteln, welche Eigenschaften der zahlreichen Controls mit bestimmten Adressen im Speicher der Control Engine verbunden sind. Die Verbindungstabelle enthält einen Eintrag pro Verbindung. Jeder Eintrag enthält die folgenden Informationen:

- **Property name:** In diesem Feld wird die Eigenschaft angegeben, der eine Variable zugeordnet ist.
- **Data source:** In diesem Feld wird die Adresse im Speicher der Control Engine für die Verbindung angegeben.
- **Update rate:** In diesem Feld wird die Aktualisierungsrate für die Verbindung definiert. Enthält das Feld keinen Wert, verwendet das Data Control die voreingestellte Aktualisierungsrate (hierbei handelt es sich um den in der Eigenschaft DefaultUpdateRate gespeicherten Wert).
- **Dead band:** In diesem Feld wird die Totzone zum automatischen Schreiben in die Control Engine oder in das Control definiert. Enthält das Feld keinen Wert, verwendet das Data Control die voreingestellte Totzone (hierbei handelt es sich um den in der Eigenschaft DefaultDeadBand gespeicherten Wert).

Wenn Sie das Data Control im Dialogfeld "Eigenschaften" einrichten, erstellt das Data Control automatisch eine Verbindungstabelle. Sie können auch ein Programm schreiben, um manuell eine Verbindungstabelle zu erstellen. Im Abschnitt 5.8 finden Sie ein Beispielprogramm zum manuellen Erstellen einer Verbindungstabelle.

5.8 Beispielprogramm zum Erstellen einer Verbindungstabelle und einer Ereignistabelle

Sie können ein Programm schreiben, das eine Tabelle erstellt, in der die Verbindungen (zugeordnete Variablen) oder Ereignisse für die Control Engine definiert werden.

Arbeiten mit einer Verbindungstabelle

Sie können eine Verbindungstabelle erstellen, um eine Variable in der Control Engine einem bestimmten Control zuzuordnen. Die Verbindungstabelle entspricht dem Register "Connections" im Dialogfeld "Eigenschaften" des Data Control.

Für jedes Element in der Verbindungstabelle müssen Sie die Eigenschaft des Control für die Verbindung, die Quelle (Adresse der zugeordneten Variablen in der Control Engine), die Aktualisierungsrate und die Totzeit definieren. Wenn Sie Verbindungen programmatisch über eine Verbindungstabelle ändern möchten, müssen Sie zuerst die Verbindung des Data Control trennen (alle Verbindungen trennen), bevor Sie Verbindungen erneut zuweisen und das Data Control wieder verbinden können.

Hinweis

Anstatt mit einer Verbindungstabelle zu arbeiten, können Sie auch die Methoden zum Lesen und Schreiben für das Data Control einsetzen (ReadVariable, ReadMultipleVariables, WriteVariable und WriteMultipleVariables). Mit Hilfe dieser Methoden können Sie über nur eine Zeile Code auf mehr Daten zugreifen.

Die Tabelle 5-1 zeigt ein Beispiel für Code in Visual Basic für ein Label Control mit dem Namen lblChange in Ihrem Arbeitsblatt für MW2 in der Control Engine. Der in MW2 gespeicherte Wert wird als Titel im Label Control angezeigt.

Tabelle 5-1 Beispielprogramm zum manuellen Erstellen einer Verbindungstabelle

Visual Basic Code	
Dim ControlTable (4) As String	
'Verbindungstabelle für lblChange definieren ControlTable (0) = "Caption" 'Eigenschaft ControlTable (1) = "MW2:WORD" 'Quelle (Adresse) ControlTable (2) = "100" 'Aktualisierungsgeschwindigkeit ControlTable (3) = "0.0" 'Totzeit	
'Verbindungstabelle mit S7Data1 verknüpfen' S7Data1.ConnectObject lblChange, ControlTable	
'Mit Control Engine verbinden S7Data1.Connect	'Mit Control Engine verbunden

Arbeiten mit einer Ereignistabelle

Sie können auch eine Ereignistabelle erstellen, um Ereignisse für die Control Engine zu definieren. Die Ereignistabelle entspricht dem Register "Events" im Dialogfeld "Eigenschaften" des Data Control. Die Tabelle 5-2 zeigt ein Beispiel für Code in Visual Basic zum Erstellen einer Ereignistabelle.

Im Abschnitt 5.9 finden Sie ein Beispielprogramm, das auf Ereignisse reagiert. Dieses Beispielprogramm definiert in einer Ereignistabelle die Ereignisse für die Control Engine.

Tabelle 5-2 Beispielprogramm zum manuellen Erstellen einer Ereignistabelle

Visual Basic Code	
Dim controlTable(4) AS String	
'Define the event keys	
ControlTable(0)="M0_0"	'Event Name
ControlTable(1)="M0.0"	'Process Variable
ControlTable(2)="500"	'Update Rate
ControlTable(3)="0"	'DeadBand
IResult=S7Data1.ConnectName("", ControlTable)	
ControlTable(0)="M0_1"	'Event Name
ControlTable(1)="M0.1"	'Process Variable
ControlTable(2)="500"	'Update Rate
ControlTable(3)="0"	'DeadBand
IResult=S7Data1.ConnectName("", ControlTable)	
ControlTable(0)="M0_2"	'Event Name
ControlTable(1)="M0.2"	'Process Variable
ControlTable(2)="500"	'Update Rate
ControlTable(3)="0"	'DeadBand
IResult=S7Data1.ConnectName("", ControlTable)	
ControlTable(0)="M0_3"	'Event Name
ControlTable(1)="M0.3"	'Process Variable
ControlTable(2)="500"	'Update Rate
ControlTable(3)="0"	'DeadBand
IResult=S7Data1.ConnectName("", ControlTable)	
ControlTable(0)="M0_4"	'Event Name
ControlTable(1)="M0.4"	'Process Variable
ControlTable(2)="500"	'Update Rate
ControlTable(3)="0"	'DeadBand
IResult=S7Data1.ConnectName("", ControlTable)	
ControlTable(0)="M0_5"	'Event Name
ControlTable(1)="M0.5"	'Process Variable
ControlTable(2)="500"	'Update Rate
ControlTable(3)="0"	'DeadBand
IResult=S7Data1.ConnectName("", ControlTable)	
ControlTable(0)="M0_6"	'Event Name
ControlTable(1)="M0.6"	'Process Variable
ControlTable(2)="500"	'Update Rate
ControlTable(3)="0"	'DeadBand
IResult=S7Data1.ConnectName("", ControlTable)	
ControlTable(0)="M0_7"	'Event Name
ControlTable(1)="M0.7"	'Process Variable
ControlTable(2)="500"	'Update Rate
ControlTable(3)="0"	'DeadBand
IResult=S7Data1.ConnectName("", ControlTable)	
End	

5.9 Beispielprogramm zum Reagieren auf Ereignisse

Sie können ein Programm schreiben, das auf Ereignisse in der Control Engine reagiert. In diesem Beispielprogramm entsprechen acht Leuchten den acht Ereignissen, die in einer Verbindungstabelle definiert sind (siehe Bild 5-13). Die Ereignisse sind mit dem Zustand der Adresse MB0 verknüpft: ein Signalzustandswechsel in MB0 erzeugt eine Reihe von Ereignissen (die für jedes Bit im Byte einen Namen haben).

Wie Sie in Bild 5-13 sehen, enthält das Programm auch folgende Elemente:

- Ein Data Control (S7Data4) für die Verbindung zur Control Engine
- Eine Zeit (Timer1), die den in MB0 gespeicherten Wert inkrementiert (wodurch dann die Control Engine die Ereignisse erzeugt)
- Eine Befehlsschaltfläche (cmdStartEvent) zum Starten oder Stoppen der Zeit (also zum Starten und Stoppen der Ereignisgenerierung)

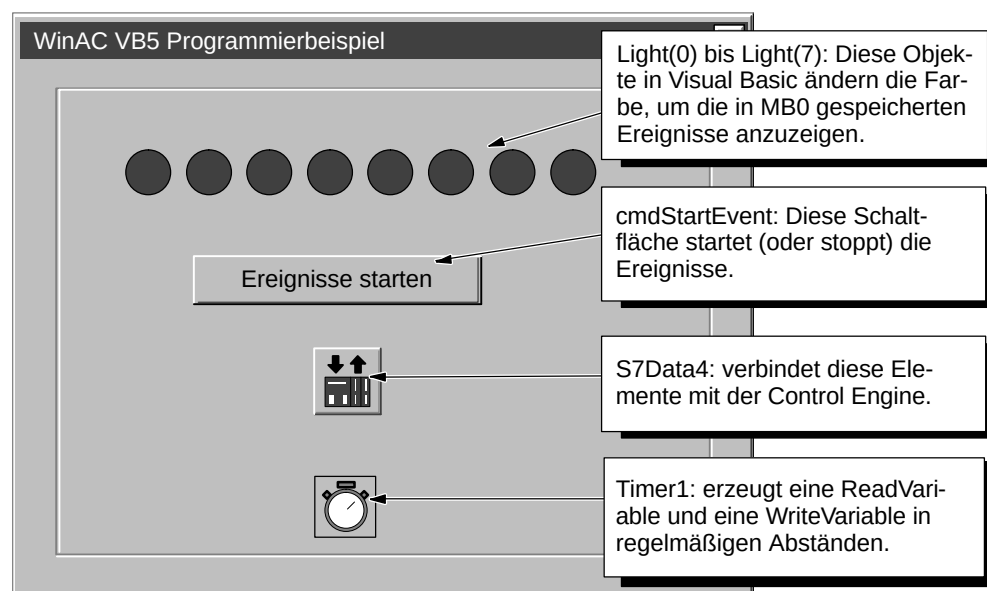


Bild 5-13 Beispielprogramm zum Reagieren auf Ereignisse in der Control Engine



Vorsicht

Wenn Sie die Zeiten in Ihrem Programm nicht deaktivieren, kann es sein, daß zeitgesteuerte Verbindungen bestehen bleiben, die weiterhin Daten in die Control Engine schreiben. Dadurch ist unvorhersehbarer Betrieb der Control Engine möglich, der zu Sachschaden oder Körperverletzung führen kann.

Damit sichergestellt ist, daß alle Verbindungen getrennt sind, wenn Ihr Programm geschlossen wird, müssen Sie im Unterprogramm Form_Unload alle Zeiten vor dem Ausdruck End deaktivieren.

Erstellen einer Verbindungstabelle zum Reagieren auf Ereignisse

Ihr Programm kann eine Ereignistabelle erstellen, um bestimmte Ereignisse in der Control Engine zu definieren. Tabelle 5-3 zeigt den Code zum Erstellen einer Verbindungstabelle, in der Ereignisschlüssel für eine Control Engine definiert werden.

Tabelle 5-3 Beispielprogramm zum Erstellen einer Verbindungstabelle zum Reagieren auf Ereignisse

Visual Basic Code	
Dim controlTable(4) AS String	
'Define the event keys	
ControlTable(0)="M0_0"	'Event Name
ControlTable(1)="M0.0"	'Process Variable
ControlTable(2)="500"	'Update Rate
ControlTable(3)="0"	'DeadBand
IResult=S7Data1.ConnectName("", ControlTable)	
ControlTable(0)="M0_1"	'Event Name
ControlTable(1)="M0.1"	'Process Variable
ControlTable(2)="500"	'Update Rate
ControlTable(3)="0"	'DeadBand
IResult=S7Data1.ConnectName("", ControlTable)	
ControlTable(0)="M0_2"	'Event Name
ControlTable(1)="M0.2"	'Process Variable
ControlTable(2)="500"	'Update Rate
ControlTable(3)="0"	'DeadBand
IResult=S7Data1.ConnectName("", ControlTable)	
ControlTable(0)="M0_3"	'Event Name
ControlTable(1)="M0.3"	'Process Variable
ControlTable(2)="500"	'Update Rate
ControlTable(3)="0"	'DeadBand
IResult=S7Data1.ConnectName("", ControlTable)	
ControlTable(0)="M0_4"	'Event Name
ControlTable(1)="M0.4"	'Process Variable
ControlTable(2)="500"	'Update Rate
ControlTable(3)="0"	'DeadBand
IResult=S7Data1.ConnectName("", ControlTable)	
ControlTable(0)="M0_5"	'Event Name
ControlTable(1)="M0.5"	'Process Variable
ControlTable(2)="500"	'Update Rate
ControlTable(3)="0"	'DeadBand
IResult=S7Data1.ConnectName("", ControlTable)	
ControlTable(0)="M0_6"	'Event Name
ControlTable(1)="M0.6"	'Process Variable
ControlTable(2)="500"	'Update Rate
ControlTable(3)="0"	'DeadBand
IResult=S7Data1.ConnectName("", ControlTable)	
ControlTable(0)="M0_7"	'Event Name
ControlTable(1)="M0.7"	'Process Variable
ControlTable(2)="500"	'Update Rate
ControlTable(3)="0"	'DeadBand
IResult=S7Data1.ConnectName("", ControlTable)	
End	

Reagieren auf die vom Beispielprogramm generierten Ereignisse

Tabelle 5-4 zeigt ein Beispiel für Code in Visual Basic, um auf verschiedene Ereignisse der Control Engine zu reagieren.

Tabelle 5-4 Beispielprogramm zum Reagieren auf Ereignisse in der Control Engine

Visual Basic Code	
<pre>Private Sub S7Data4_ValueChanged(ByVal Property As String, ByVal VarName As String, ByVal Value As Variant, ByVal Quality As Integer) 'Wertet aus, welches Ereignis aufgetreten ist Select Case Property</pre>	
<pre> Case "M0_0" If Value = True Then Light(0).FillColor = vbGreen Else Light(0).FillColor = vbRed End If</pre>	'Ereignis M0_0 schaltet Light(0) grün ein
<pre> Case "M0_1" If Value = True Then Light(1).FillColor = vbGreen Else Light(1).FillColor = vbRed End If</pre>	'Ereignis M0_1 schaltet Light(1) grün ein
<pre> Case "M0_2" If Value = True Then Light(2).FillColor = vbGreen Else Light(2).FillColor = vbRed End If</pre>	'Ereignis M0_2 schaltet Light(2) grün ein
<pre> Case "M0_3" If Value = True Then Light(3).FillColor = vbGreen Else Light(3).FillColor = vbRed End If</pre>	'Ereignis M0_3 schaltet Light(3) grün ein
<pre> Case "M0_4" If Value = True Then Light(4).FillColor = vbGreen Else Light(4).FillColor = vbRed End If</pre>	'Ereignis M0_4 schaltet Light(4) grün ein
<pre> Case "M0_5" If Value = True Then Light(5).FillColor = vbGreen Else Light(5).FillColor = vbRed End If</pre>	'Ereignis M0_5 schaltet Light(5) grün ein
<pre> Case "M0_6" If Value = True Then Light(6).FillColor = vbGreen Else Light(6).FillColor = vbRed End If</pre>	'Ereignis M0_6 schaltet Light(6) grün ein

Tabelle 5-4 Beispielprogramm zum Reagieren auf Ereignisse in der Control Engine, continued

<pre>Case "M0_7" If Value = True Then Light(7).FillColor = vbGreen Else Light(7).FillColor = vbRed End If End Select End Sub</pre>	<pre>'Ereignis M0_7 schaltet Light(7) grün ein</pre>
--	--

Ausführen des Beispielprogramms (Erzeugen der Ereignisse in der Control Engine)

Tabelle 5-5 zeigt ein Beispiel für Code in Visual Basic zum Ändern des in MB0 gespeicherten Werts. Wird der Wert in MB0 geändert, erzeugt die Control Engine die Ereignisse, die in der Verbindungstabelle (Tabelle 5-3) definiert sind.

- Die Schaltfläche (cmdStartEvents) startet oder stoppt die Zeit (Timer1).
- Die Zeit (Timer1) liest den in MB0 gespeicherten Wert der Control Engine, inkrementiert den Wert und schreibt den neuen Wert zurück in die Control Engine.

Der geänderte Wert in MB0 verursacht, daß die Control Engine die Ereignisse erzeugt.



Vorsicht

Wenn Sie die Zeiten in Ihrem Programm nicht deaktivieren, kann es sein, daß zeitgesteuerte Verbindungen bestehen bleiben, die weiterhin Daten in die Control Engine schreiben. Dadurch ist unvorhersehbarer Betrieb der Control Engine möglich, der zu Sachschaden oder Körperverletzung führen kann.

Damit sichergestellt ist, daß alle Verbindungen getrennt sind, wenn Ihr Programm geschlossen wird, müssen Sie im Unterprogramm Form_Unload alle Zeiten vor dem Ausdruck End deaktivieren.

Tabelle 5-5 Weitere Unterprogramme zum Ausführen des Beispielprogramms

Visual Basic Code

```
Private Sub cmdStartEvents_Click()  
    If cmdStartEvents.Caption = "Ereignisse starten" Then  
        Timer1.Enabled = True  
        cmdStartEvents.Caption = "Ereignisse stoppen"  
    Else  
        Timer1.Enabled = False  
        cmdStartEvents.Caption = "Ereignisse starten"  
    End If  
End Sub
```

```
Private Sub Timer1_Timer()  
    Dim mb0 As Variant  
    Dim my_state As Long  
    S7Data4.ReadVariable "MB0", mb0, my_state, 0  
    If mb0 < 254 Then  
        mb0 = mb0 + 1  
    Else  
        mb0 = 0  
    End If  
    Label2.Caption = mb0  
    S7Data4.WriteVariable "MB0", mb0, 0  
End Sub
```

5.10 Beispielprogramme zum Lesen und Schreiben von Daten

Sie können ein Programm schreiben, das den Zugriff (lesend oder schreibend) auf Daten in der Control Engine initiiert.

Sie können einzelne Variablen, mehrere Variablen oder Felder von Variablen lesen oder schreiben. (Zum Lesen und Schreiben von booleschen Daten müssen Sie die Methoden ReadMultiVariables und WriteMultiVariables einsetzen, siehe Abschnitt 5.11).

Informationen zu den Speicherbereichen von S7-Automatisierungssystemen entnehmen Sie Anhang A.



Vorsicht

Wenn Sie die Zeitfunktion oder Haltepunkte in Visual Basic mit SIMATIC Computing verwenden, kann dies zu Fehlern führen, die evt. Ihren Computer bzw. die Anwendung zum Absturz bringen bzw. verriegeln können. Je nach Konfiguration kann die Anwendung dadurch die Kommunikation mit der Control Engine beenden. Dies kann unvorhersehbare Reaktionen im Prozeß auslösen, die zu Tod, schwerer Körperverletzung und/oder Sachschaden führen können.

Installieren Sie einen physikalischen NOT-AUS-Schaltkreis für die Maschine bzw. den Prozeß.

Timer in VB: Bei der Timer-Funktion in Visual Basic Version 5 kann ein Timer den in der gleichen Einweg-Bearbeitung befindlichen Code unterbrechen, wodurch Fehler mit schwerwiegenden Folgen auftreten können. Beachten Sie folgende Richtlinien, wenn Sie VB-Timer mit SIMATIC Computing einsetzen:

- Deaktivieren Sie die Timer immer im Unterprogramm Form_Unload. Andernfalls kann ein Timer ein Ereignis auslösen, während das VB-Programm geschlossen wird. Diese Bedingung kann einen Absturz bzw. eine Verriegelung Ihres Computers oder der Anwendung herbeiführen. Es kann auch sein, daß die Anwendung unsichtbar weiterläuft.
 - Wenn Sie den Timer im Unterprogramm Form_Load starten, kann es sein, daß das Zeitereignis auftritt, bevor die anderen Objekte instanziiert wurden. Um sicherzustellen, daß die Objekte korrekt instanziiert werden, starten Sie einen Timer im Unterprogramm Form_Load immer mit einem großen Intervall (z.B. 1 oder 2 Sekunden). Nachfolgende Timer-Intervalle können kürzer sein.
-

Lesen einer einzelnen Variablen in der Control Engine

Tabelle 5-6 zeigt ein Beispiel für Code in Visual Basic zum Einsatz der Methode ReadVariable des Data Control zum Lesen einer einzelnen Variablen in der Control Engine.

Tabelle 5-6 Lesen einer einzelnen Variablen in der Control Engine

Visual Basic Code
<pre> Private Sub ReadSingleRealVariable Dim rc As Long Dim name_s As String Dim value_v As Variant Dim state_l As Long Dim timeout_l As Long 'Realen Wert (Gleitpunkt) lesen name_s = "MD0:REAL" timeout_l = 0 rc = S7Data3.ReadVariable(name_s, value_v, state_l, timeout_l) 'Wert und Return-Code in einem Listenfeld anzeigen ListBox1.Clear ListBox1.AddItem "RetCode = " & Hex(rc) ListBox1.AddItem " - " & name_s & " = " & value_v ListBox1.AddItem " - State = " & Hex(state_l) End Sub </pre>

Schreiben einer einzelnen Variablen in die Control Engine

Tabelle 5-7 zeigt ein Beispiel für Code in Visual Basic zum Einsatz der Methode WriteVariable des Data Control zum Schreiben einer einzelnen Variablen in die Control Engine.

Tabelle 5-7 Schreiben einer einzelnen Variablen in die Control Engine

Visual Basic Code
<pre> Private Sub WriteSingleRealVariable Dim rc As Long Dim name_s As String Dim value_v As Variant Dim timeout_l As Long 'Realen Wert (Gleitpunkt) schreiben name_s = "MD0:REAL" value_v = (Rnd * 1000) timeout_l = 100 rc = S7Data3.WriteVariable(name_s, value_v, timeout_l) 'Wert und Return-Code in einem Listenfeld anzeigen ListBox1.Clear ListBox1.AddItem "Wrote " & name_s & " = " & value_v ListBox1.AddItem "Return Code = " & Hex(rc) End Sub </pre>

Lesen eines Felds in der Control Engine

Tabelle 5-8 zeigt ein Beispiel für Code in Visual Basic zum Einsatz der Methode ReadVariable des Data Control zum Lesen eines Felds aus Daten in der Control Engine.

Hinweis

Lesen und schreiben Sie Daten vom Datentyp STRING und CHAR als Visual Basic-Daten BSTR. Verwenden Sie kein Feld vom Datentyp CHAR, um den Datentyp STRING zu emulieren.

Verwenden Sie ein BSTR für jeden STRING bzw. jedes CHAR, unabhängig von der Länge der Daten, auf die zugegriffen wird. Beispiel:

- Zum Zugreifen auf CHAR[50] (das 50 Bytes bzw. 50 Zeichen angibt), verwenden Sie ein BSTR von bis zu 50 Bytes und **nicht** 50 einzelne BSTRs.
 - Zum Zugreifen auf STRING[50] (das eine Zeichenfolge von 50 Zeichen angibt), verwenden Sie ein BSTR von bis zu 50 Bytes und **nicht** 50 BSTRs.
-

Tabelle 5-8 Beispielprogramm zum Lesen eines Felds aus Variablen

Visual Basic Code

```
Private Sub ReadArrayOfReals
Dim rc As Long
Dim name_s As String
Dim value_v As Variant
Dim state_1 As Long
Dim timeout_l As Long
Randomize

'Feld aus realen Werten (Gleitpunkt) lesen
name_s = "MD0:Real[3]"
timeout_l = 0

rc = S7Data3.ReadVariable(name_s, value_v, state_1, timeout_l)

'Werte und Return-Codes für das Feld in einem Listenfeld anzeigen
ListBox1.Clear
ListBox1.AddItem "Return Code = " & Hex(rc)
ListBox1.AddItem "  - name_s & " = " & value_v(0) & " " & value_v(1) & " " & value_v(2)
ListBox1.AddItem "  - State = " & Hex(state_1)

End Sub
```

Schreiben eines Felds in die Control Engine

Tabelle 5-8 zeigt ein Beispiel für Code in Visual Basic zum Einsatz der Methode WriteVariable des Data Control zum Schreiben eines Felds aus Daten in die Control Engine.

Hinweis

Lesen und schreiben Sie Daten vom Datentyp STRING und CHAR als Visual Basic-Daten BSTR. Verwenden Sie kein Feld vom Datentyp CHAR, um den Datentyp STRING zu emulieren.

Verwenden Sie ein BSTR für jeden STRING bzw. jedes CHAR, unabhängig von der Länge der Daten, auf die zugegriffen wird. Beispiel:

- Zum Zugreifen auf CHAR[50] (das 50 Bytes bzw. 50 Zeichen angibt), verwenden Sie ein BSTR von bis zu 50 Bytes und **nicht** 50 einzelne BSTRs.
- Zum Zugreifen auf STRING[50] (das eine Zeichenfolge von 50 Zeichen angibt), verwenden Sie ein BSTR von bis zu 50 Bytes und **nicht** 50 BSTRs.

Tabelle 5-9 Beispielprogramm zum Schreiben eines Felds aus Variablen

Visual Basic Code

```
Private Sub WriteArrayOfReals
Dim rc As Long
Dim name_s As String
Dim timeout_l As Long
Dim value_b(2) As Byte      ' Byte schreiben
Dim value_w(2) As Integer   ' Wort schreiben
Dim value_r(2) As Single    ' Realzahl schreiben

'Feld aus realen Werten (Gleitpunkt) lesen
name_s = "MD0:REAL[3]"
value_r(0) = (Rnd * 1000)
value_r(1) = (Rnd * 1000)
value_r(2) = (Rnd * 1000)
timeout_l = 100

rc = S7Data3.WriteVariable(name_s, value_r, timeout_l)

'Werte und Return-Codes für das Feld in einem Listenfeld anzeigen
ListBox1.Clear
ListBox1.AddItem "Return Code = " & Hex(rc)
ListBox1.AddItem "  - Wrote MD0:REAL[0] = " & value_r(0)
ListBox1.AddItem "  - Wrote MD0:REAL[1] = " & value_r(1)
ListBox1.AddItem "  - Wrote MD0:REAL[2] = " & value_r(2)

End Sub
```

Lesen von mehreren Variablen in der Control Engine

Tabelle 5-10 zeigt ein Beispiel für Code in Visual Basic zum Einsatz der Methode ReadMultiVariable des Data Control zum Lesen von mehreren Variablen in der Control Engine.

Tabelle 5-10 Lesen von mehreren Variablen in der Control Engine

Visual Basic Code
<pre> Private Sub ReadMultiReals Dim i As Integer Dim rc As Long Dim names_array(2) As String Dim values_v As Variant Dim states_v As Variant 'Drei reale Werte (Gleitpunkt) lesen For i = 0 To 2 names_array(i) = "MD" & i * 4 & ":REAL" Next i rc = S7Data3.ReadMultiVariables(names_array, values_v, states_v) 'Wert und Return-Code in einem Listenfeld anzeigen ListBox1.Clear ListBox1.AddItem "RetCode = " & Hex(rc) For i = 0 To 2 ListBox1.AddItem " - " & names_array(i) & " = " & values_v(i) & - vbTab & " State = " & Hex(states_v(i)) Next i End Sub </pre>

Schreiben von mehreren Variablen in die Control Engine

Tabelle 5-11 zeigt ein Beispiel für Code in Visual Basic zum Einsatz der Methode WriteMultiVariable des Data Control zum Schreiben von mehreren Variablen in die Control Engine.

Tabelle 5-11 Schreiben von mehreren Variablen in die Control Engine

Visual Basic Code
<pre> Private Sub cmdWriteMultVar_Click(Index As Integer) Dim i As Integer Dim rc As Long Dim names_array(2) As String Dim values_v(2) As Variant Dim states_v As Variant 'Drei reale Werte (Gleitpunkt) schreiben For i = 0 To 2 names_array(i) = "MD" & i * 4 & ":REAL" values_v(i) = (Rnd * 1000) Next i rc = S7Data2.WriteMultiVariables(names_array, values_v, states_v) 'Werte und Return-Codes in einem Listenfeld anzeigen lstReal.Clear lstReal.AddItem "RetCode = " & Hex(rc) For i = 0 To 2 lstReal.AddItem " - " & names_array(i) & " = " & values_v(i) & vbTab & - " State = " & Hex(states_v(i)) Next i End Sub </pre>

5.11 Beispielprogramm zum Lesen und Schreiben von booleschen Daten

Zum Lesen und Schreiben von booleschen Daten müssen Sie die Methoden ReadMultiVariables und WriteMultiVariables einsetzen. Tabelle 5-12 zeigt ein Beispielprogramm zum Lesen und Schreiben von Feldern aus booleschen Daten.

Tabelle 5-12 Lesen und Schreiben von mehreren Variablen

Visual Basic Code

```
Private Sub Read_Booleans()
    Dim mybools(7) As String
    Dim vals_v As Variant
    Dim states_v As Variant
    Dim rc As Long

    mybools(0) = "m0.0"
    mybools(1) = "m0.1"
    mybools(2) = "m0.2"
    mybools(3) = "m0.3"
    mybools(4) = "m0.4"
    mybools(5) = "m0.5"
    mybools(6) = "m0.6"
    mybools(7) = "m0.7"

    rc = S7Data1.ReadMultiVariables(mybools, vals_v, states_v)
End Sub

Private Sub Write_Booleans()
    Dim mybools(7) As String
    Dim myvals(7) As Variant
    Dim states_v As Variant
    Dim rc As Long
    mybools(0) = "m0.0"
    mybools(1) = "m0.1"
    mybools(2) = "m0.2"
    mybools(3) = "m0.3"
    mybools(4) = "m0.4"
    mybools(5) = "m0.5"
    mybools(6) = "m0.6"
    mybools(7) = "m0.7"

    myvals(0) = False
    myvals(1) = False
    myvals(2) = False
    myvals(3) = False
    myvals(4) = False
    myvals(5) = False
    myvals(6) = False
    myvals(7) = False

    rc = S7Data1.WriteMultiVariables(mybools, myvals, states_v)
End Sub
```

5.12 Eigenschaften, Methoden und Ereignisse des Data Control

Mit den in Tabelle 5-13 aufgeführten Eigenschaften und Methoden können Sie das Data Control bearbeiten.

Tabelle 5-13 Eigenschaften und Methoden des Data Control

Eigenschaft / Methode	Beschreibung	Seite
Eigenschaft Activated	Gibt an, ob alle Verbindungen aktiviert sind oder nicht.	B-1
Eigenschaft AutoConnect	Gibt an, ob die konfigurierten Verbindungen zur Laufzeit aufgebaut werden oder nicht.	B-3
Eigenschaft AutoConnectTimeout	Gibt einen Timeout-Wert an.	B-4
Methode Connect	Baut alle konfigurierten Verbindungen auf.	B-7
Methode ConnectName	Baut für ein Objekt, dessen Name angegeben wird, die Verbindungen auf.	B-8
Methode ConnectObject	Baut für ein angegebenes Objekt die Verbindungen auf.	B-9
Eigenschaft ControlEngine	Gibt die Control Engine für die Verbindung an.	
Eigenschaft DefaultDeadband	Gibt die Totzeit an, die das SIMATIC Data Control verwendet, wenn in der Verbindungstabelle keine Totzeit definiert ist.	B-12
Eigenschaft DefaultUpdateRate	Gibt die Aktualisierungsgeschwindigkeit an, die das Data Control verwendet, wenn in der Verbindungstabelle keine Aktualisierungsgeschwindigkeit definiert ist.	B-13
Methode Disconnect	Trennt alle aufgebauten Verbindungen.	B-14
Eigenschaft MultipleEngines	Gibt an, ob die Verbindung zu einer bestimmten oder zu mehreren Control Engines besteht.	B-27
Eigenschaft PCName	Gibt die Netzkennung für einen entfernten Computer an (für Verbindungen im Netz).	B-28
Methode PropertyChangedName	Informiert das SIMATIC Data Control, daß sich der Wert einer Eigenschaft eines verbundenen Control, das durch "Name" angegeben wird, geändert hat.	B-30
Methode PropertyChangedObject	Informiert das SIMATIC Data Control, daß sich der Wert einer Eigenschaft eines verbundenen Control, das durch "Object" angegeben wird, geändert hat.	B-31
Methode ReadMultiVariables	Liest den Zustand mehrerer Variablen in der Control Engine.	B-33
Methode ReadVariable	Liest den Zustand einer bestimmten Variablen in der Control Engine.	B-33
Eigenschaft ShowErrorBoxes	Gibt an, ob bei Fehlern, die vom Anwender hervorgerufen werden, die Standard-Fehlermeldungen angezeigt werden sollen.	B-36
Eigenschaft TagSource	Gibt die Quelle (z.B. eine Symboldatei) der symbolischen Informationen an, die verwendet werden soll, wenn Variablen zugeordnet und Control Engines gekennzeichnet werden sollen.	B-39
Methode WriteMultiVariables	Schreibt neue Werte in verschiedene Variablen in der Control Engine.	B-44
Methode WriteVariable	Schreibt einen neuen Wert in eine bestimmte Variable in der Control Engine.	B-45

Das Data Control reagiert auf die in Tabelle 5-14 aufgeführten Ereignisse.

Tabelle 5-14 Ereignisse des Data Control

Ereignis	Beschreibung	Seite
ConnectionError	Tritt auf, wenn in einer Verbindung ein Fehler auftritt.	C-1
ValueChanged	Tritt auf, wenn sich der Wert einer verbundenen Variablen ändert und für den Aufruf der Methode Connect kein verbundenes Ereignis angegeben war.	C-9

Fehlercodes für das Data Control (ConnectionError Event)

Tritt im Data Control ein Fehler auf, erzeugt das Control das Ereignis ConnectionError. Ihr Programm kann das Ereignis ConnectionError aufnehmen und auf bestimmte Situationen reagieren. Die Fehler, die das Ereignis ConnectionError erkennen kann, sind OLE-Fehler wie E_FAIL oder E_OUTOFMEM. Tabelle 5-15 führt einige der Fehlercodes auf.

Tabelle 5-15 Fehlercodes des Data Control

Fehlercode	Beschreibung
0 x 80004005	Allgemeiner OLE-Fehler
0x8007000E	Nicht genügend freier Speicher
0 x 80070057	Ungültige Syntax der Variablen
0xC0040004	Ungültiger oder unbekannter Datentyp
0xC0040007	Ungültiger Variablenart
0xC0040008	Ungültige Syntax für die Definition des Elements
0xC004000B	An WRITE übergebener Wert ist außerhalb des Bereichs

Kapitelübersicht

SIMATIC Computing verfügt über ActiveX User Controls für den Zugriff auf Prozeßdaten. Im Dialogfeld "Eigenschaften" des Data Control stellen Sie die Verbindung zwischen dem User Control und der Control Engine her. Für jedes Control gibt es ein Dialogfeld "Eigenschaften", in dem Sie die Funktionsweise des Control einstellen können.

- Mit dem Button Control können Sie einzelne Bits im Speicher ein- und ausschalten.
- Das Edit Control bietet Zugriff auf die Adressen im Speicher der Control Engine.
- Mit dem Label Control können Sie eine Konstante als Zeichenfolge anzeigen.
- Das Slider Control dient als Schnittstelle zum Überwachen und Ändern von Analogvariablen.

Kapitel	Beschreibung	Seite
6.1	Verbinden der User Controls mit Prozeßdaten	6-2
6.2	Arbeiten mit dem Eigenschaftsdialog des Button Control	6-4
6.3	Eigenschaften und Methoden des Button Control	6-9
6.4	Ereignisse des Button Control	6-10
6.5	Arbeiten mit dem Eigenschaftsdialog des Edit Control	6-11
6.6	Eigenschaften und Methoden des Edit Control	6-18
6.7	Ereignisse des Edit Control	6-19
6.8	Fehlercodes des Edit Control	6-20
6-21	Arbeiten mit dem Eigenschaftsdialog des Label Control	6-21
6.10	Eigenschaften und Methoden des Label Control	6-26
6.11	Ereignisse des Label Control	6-26
6.12	Arbeiten mit dem Eigenschaftsdialog des Slider Control	6-27
6.13	Eigenschaften und Methoden des Slider Control	6-34
6.14	Ereignisse des Slider Control	6-35

6.1 Verbinden der User Controls mit Prozeßdaten

Wenn Sie zwischen dem Button, Edit oder Slider Control und Ihren Prozeßdaten eine Verbindung herstellen möchten, ordnen Sie der Eigenschaft Value des Control eine Variable (ein Bit) zu. Wenn Sie zwischen dem Label Control und Ihren Prozeßdaten eine Verbindung herstellen möchten, ordnen Sie der Eigenschaft Caption des Label Control eine Variable (ein Bit) zu. Die Variable kann nicht im Dialogfeld "Eigenschaften" für das Control zugeordnet werden. Sie müssen stattdessen das Dialogfeld "Eigenschaften" für das Data Control aufrufen und die Schaltfläche im Register "Connections" in der Liste der Controls auswählen (siehe Bild 6-1).

Zum Einstellen aller anderen Eigenschaften mit Ausnahme der Eigenschaft Value können Sie im Dialogfeld "Eigenschaften" für das Control arbeiten. Öffnen Sie das Menü **Bearbeiten** bzw. klicken Sie mit der rechten Maustaste und wählen Sie den Befehl **Eigenschaften** für das Control.

Hinweis

Wenn Sie das Control mit den tatsächlichen Prozeßdaten verbinden möchten, müssen Sie eine Verbindung über das Data Control herstellen.

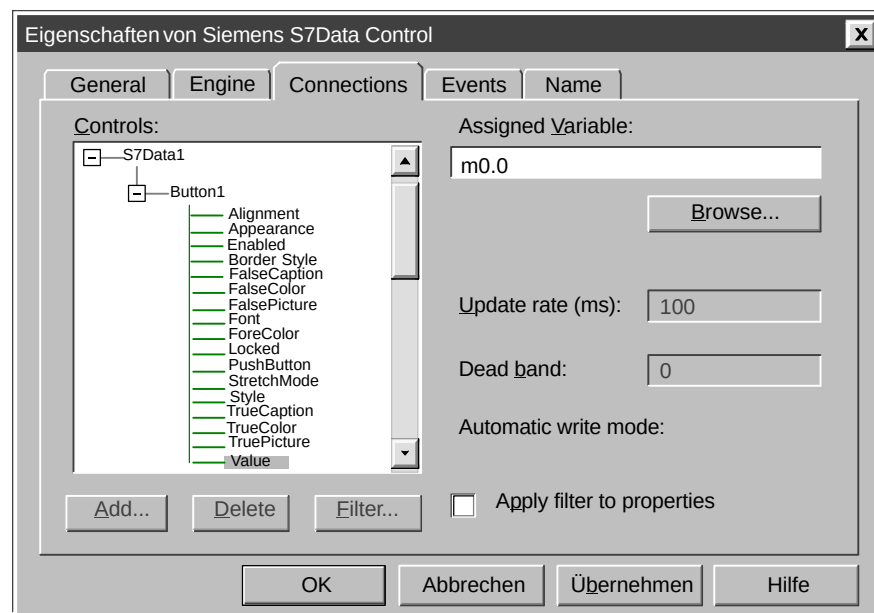


Bild 6-1 Zuordnen von Variablen für das Button oder Edit Control

Angeben von Variablen und Datentypen

Sie können in SIMATIC Computing für eine Variable, die Sie einer Eigenschaft eines Control zuordnen, einen Datentyp angeben. Beachten Sie dabei, daß Sie dem Button Control nur einzelne Bits in der Control Engine zuordnen können. Der einzige gültige Datentyp für ein Button Control ist BOOL.

Alternativ können Sie in SIMATIC Computing den Datentyp angeben, wenn Sie einer Eigenschaft eines SIMATIC Controls eine Variable zuordnen. Sie definieren den Datentyp, indem Sie die absolute Adresse im Speicher angeben, dann einen Doppelpunkt (:) gefolgt von dem Datentyp. Gehen Sie beim Zuordnen von Datentypen vorsichtig vor. Wenn Sie eine Verbindung zu einem Edit Control herstellen, werden die Werte einiger S7-Datentypen nicht korrekt angezeigt, wenn der zugeordnete Datentyp nicht dem Wert im Feld "Data Format" im Dialogfeld "Eigenschaften" des Edit Control entspricht. Sie können beispielsweise eine zugeordnete Variable als Datentyp REAL definieren, indem Sie beim Zuordnen der Variablen "MD100:real" eingeben; Sie müssen dann jedoch im Feld "Data Format" im Dialogfeld "Eigenschaften" des Edit Control den Wert Real eingeben.

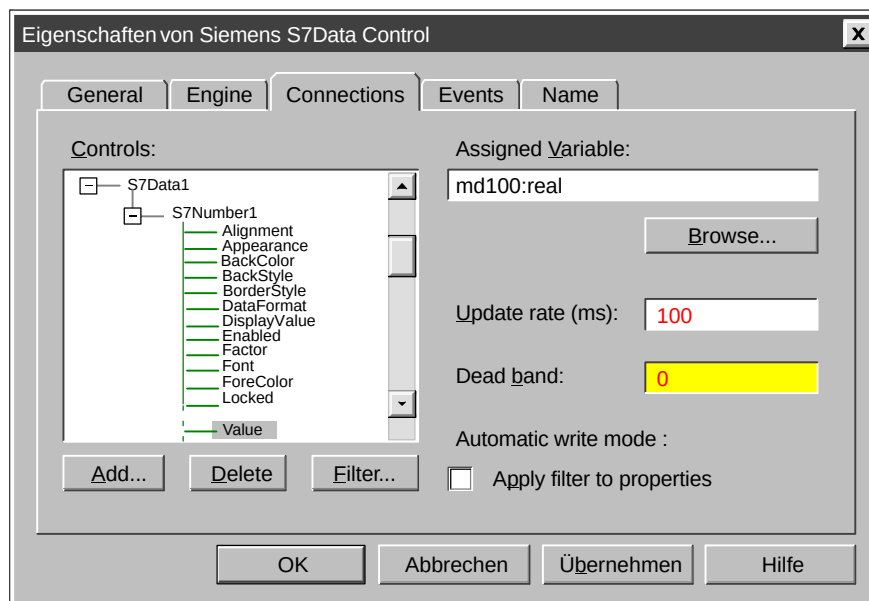


Bild 6-2 Zuordnen von Variablen für das Edit Control

6.2 Arbeiten mit dem Eigenschaftsdialog des Button Control

Mit dem Button Control können Sie der Anzeige einer Schaltfläche ein Datenbit aus Ihrem Prozeß zuordnen. Sie ordnen die Schaltfläche Ihrem Prozeß zu, indem Sie der Schaltfläche eine Variable (nämlich die gewünschte Adresse des Bit) zuweisen. Sie können dann die Schaltfläche ein- und ausschalten, um so den Zustand des Bit zu ändern. Die Farbe der Schaltfläche ändert sich automatisch, wenn sich der Zustand des Bit im Prozeß ändert.

Das Button Control bietet Zugriff auf einzelne Bits im Speicher der Control Engine und verfügt über zwei Zustände: 0 (aus) oder 1 (ein). Wenn Sie auf das Button Control klicken, werden die Daten in der Control Engine geändert.

Das Button Control liest und schreibt boolesche Werte (ein Bit).

Definieren des Titels und Freigeben des Control (im Register "General")

Im Register "General" im Dialogfeld "Eigenschaften" (siehe Bild 6-3) können Sie die beiden Titel (Caption) für das Button Control definieren:

- "Alignment" legt die Ausrichtung des Texts (links, zentriert oder rechts) fest.
- "TrueCaption": Geben Sie hier den Text ein, der im Control angezeigt werden soll, wenn das Bit "true" (bzw. "1" oder "ein") ist.
- "FalseCaption": Geben Sie hier den Text ein, der im Control angezeigt werden soll, wenn das Bit "false" (bzw. "0" oder "aus") ist.
- Die Eigenschaft "Style" stellt die Darstellungsart (standard oder graphisch) des Control ein. Die Darstellungsart "graphisch" bedeutet, daß ein Bitmap verwendet wird.
- "Appearance": Wenn Sie für diese Eigenschaft "3D" einstellen, wird das Control in dreidimensionalem Format angezeigt. (Sie müssen dann auch für die Eigenschaft "Border Style" die Option "Fixed Single" einstellen, damit das dreidimensionale Format aktiviert werden kann.) Als zweite Möglichkeit können Sie "Flat" einstellen. Es wird dann ein zweidimensionaler rechteckiger Rahmen um das Control angezeigt.
- "BorderStyle": Wenn Sie für diese Eigenschaft den Wert "Fixed Single" einstellen, wird das Control mit rechteckigem Rahmen angezeigt. Wenn Sie für die Eigenschaft "None" einstellen, wird kein Rahmen angezeigt.
- "StretchMode" gibt den Stretch-Modus des graphischen Elements des Control an.
- Das Kontrollkästchen "Enabled" legt fest, ob das Button Control auf Ereignisse reagiert. Es erzeugt im deaktivierten Zustand keine Ereignisse. Als Voreinstellung ist dieses Kontrollkästchen aktiviert (durch ein Häkchen gekennzeichnet).
- Das Kontrollkästchen "Locked" legt fest, ob das Control schreibgeschützt ist. Im verriegelten Zustand können Sie keine Werte ändern.
- Die Eigenschaft "Pushbutton" gibt an, ob das Control wie eine Taste funktioniert. Die Eigenschaft legt den Betriebszustand des Control fest: Ist PushButton = True oder 1, wird der Wert True so lange erhalten, bis das Button Control gedrückt wird (Ereignis MouseDown).

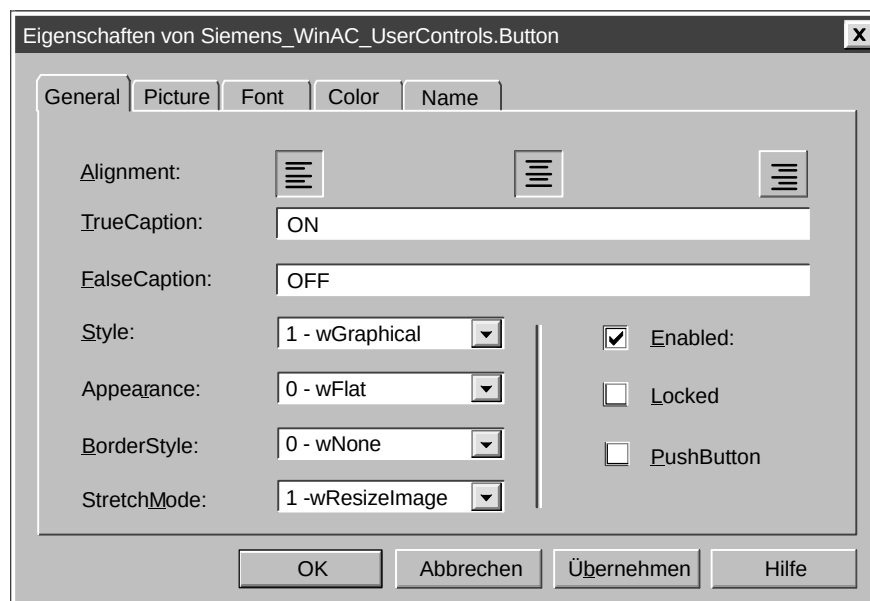


Bild 6-3 Eigenschaften des Button Control (Register "General")

Definieren der Abbildung für das Button Control (im Register "Picture")

Im Register "Picture" im Dialogfeld "Eigenschaften" (siehe Bild 6-4) können Sie ein Bild für die beiden Zustände des Button Control auswählen: Sie markieren den Zustand "Aus" (FalsePicture) oder den Zustand "Ein" (TruePicture) und wählen dann über die Schaltfläche "Browse" das Bild für den jeweiligen Zustand aus. Sie können beliebige Bilder für die Zustände "Ein" und "Aus" auswählen, doch Graphiken sind nur zulässig, wenn Sie im Register "General" im Feld "Style" den Wert "1 - wGraphical" angegeben haben. Vordefinierte Bitmaps befinden sich im Verzeichnis WinAC\WinCP\bitmaps.

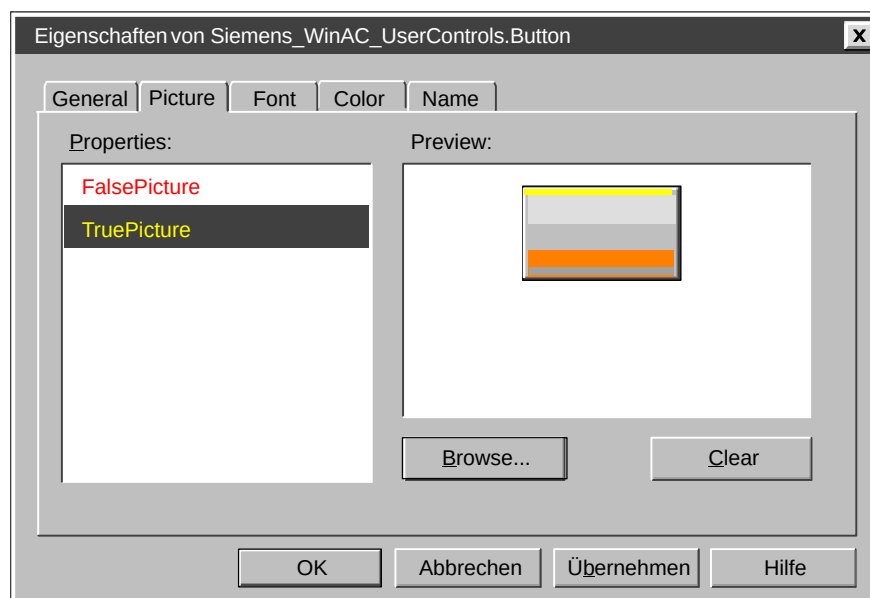


Bild 6-4 Eigenschaften des Button Control (Register "Picture")

Definieren der Schriftart für das Button Control (im Register "Font")

Im Register "Font" im Dialogfeld "Eigenschaften" (siehe Bild 6-5) können Sie die Schriftart und die Schriftgröße für den Text auf dem Button Control definieren:

- "Font": Wählen Sie hier aus einer Liste von Standard-Schriftarten eine für den Text aus.
- "Size": Wählen Sie hier die Größe in Punkt aus oder geben Sie eine bestimmte Größe in Punkt für den Text ein.
- "Effects": Wählen Sie hier noch andere typographische Optionen (fett, kursiv, unterstrichen, durchgestrichen) aus.

Das Feld "Sample Text" zeigt ein Beispiel für die in der Eigenschaft Font ausgewählte Schriftart an.

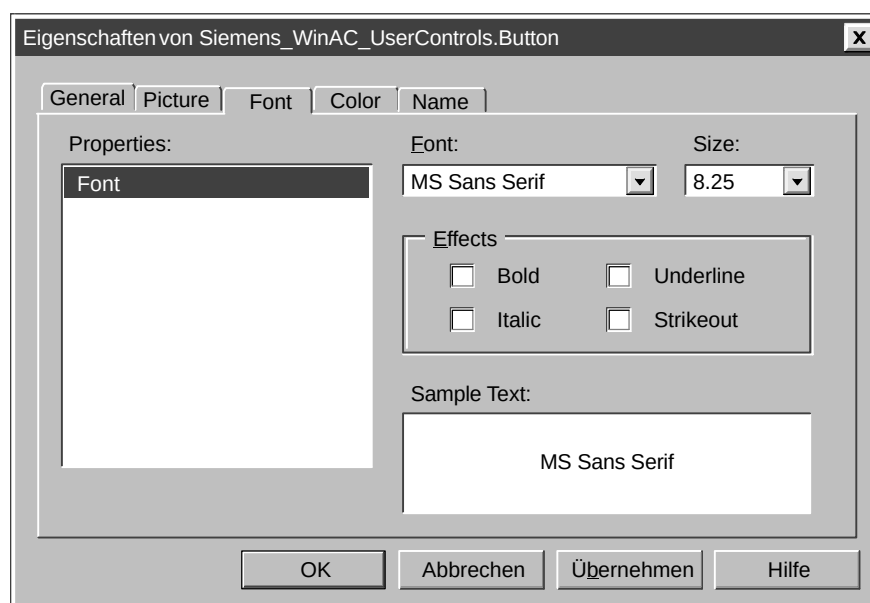


Bild 6-5 Eigenschaften des Button Control (Register "Font")

Definieren der Farbe für das Button Control (im Register "Color")

Im Register "Color" im Dialogfeld "Eigenschaften" (siehe Bild 6-6) können Sie die Farben für die beiden Zustände und für den Text des Button Control definieren: Sie können aus den Standardfarben auswählen, doch Sie können auch Ihre eigenen Farben erstellen.

- Sie markieren den Zustand "Aus" (FalseColor) oder den Zustand "Ein" (TrueColor) und wählen dann die Farbe für den jeweiligen Zustand aus der Farbpalette aus.
- Sie können auch die "ForeColor" definieren, in der der Text in einem Objekt angezeigt wird.

Hinweis

Die Eigenschaften FalseColor und TrueColor können nur geändert werden, wenn Sie im Register "General" Style: Standard ausgewählt haben. Die Eigenschaft ForeColor (Farbe des Texts) können Sie sowohl für Style: Standard und für Style: Graphical ändern.

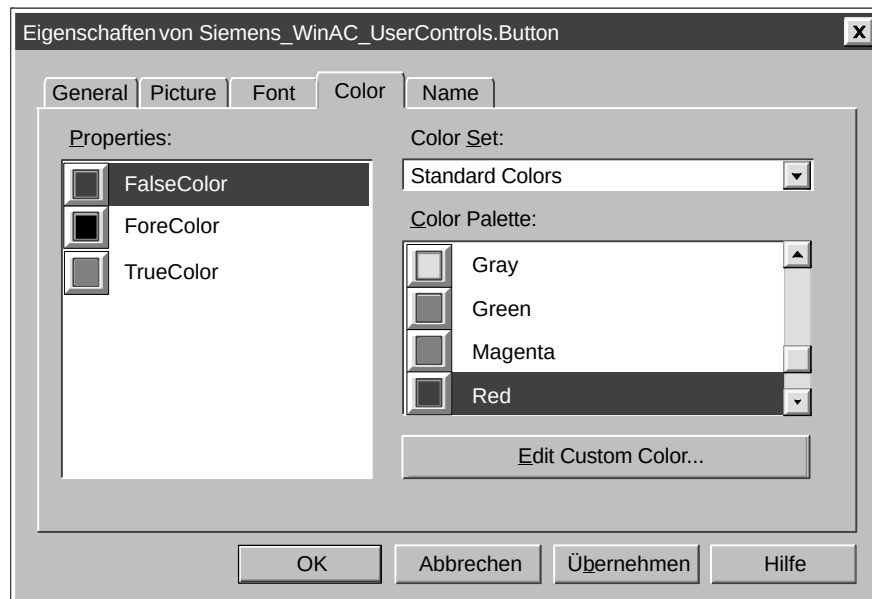


Bild 6-6 Eigenschaften des Button Control (Register "Color")

Register "Name"

Im Register "Name" des Dialogfelds "Eigenschaften" (siehe Bild 6-12) können Sie den Controls, die Sie in den Container eingefügt haben, beschreibende Namen zuordnen. Dieses Register ist nur verfügbar, wenn Sie mit Controls im SIMATIC Computing Container arbeiten.

Sie geben den neuen Namen in das Feld "Control Name" ein und bestätigen mit "Übernehmen" oder "OK". Der neue Name wird nun in der Liste "Select Control" in der Funktionsleiste aufgeführt. Durch Doppelklick auf das gewünschte Control rufen Sie die Eigenschaften des Control im Dialogfeld "Eigenschaften" auf.

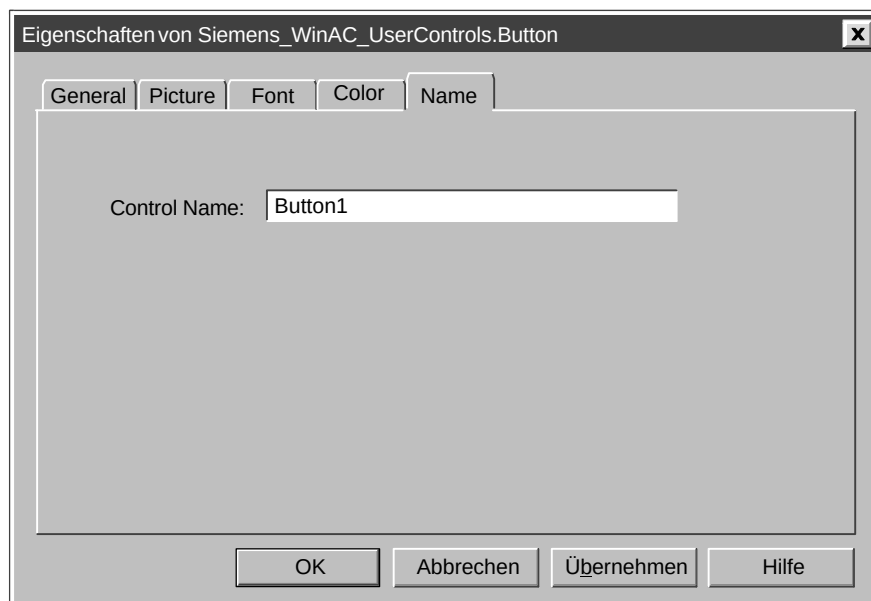


Bild 6-7 Eigenschaften des Button Control (Register "Name")

6.3 Eigenschaften und Methoden des Button Control

Mit den in Tabelle 6-1 aufgeführten Eigenschaften und Methoden können Sie das Button Control bearbeiten.

Tabelle 6-1 Eigenschaften und Methoden des Button Control

Eigenschaft / Methode	Beschreibung	Seite
Methode AboutBox	Zeigt das Dialogfeld "Info" des Control an.	B-1
Eigenschaft Alignment	Legt die Ausrichtung des Texts fest.	B-2
Eigenschaft Appearance	Legt fest, ob das Control mit 3D-Effekt angezeigt wird.	B-3
Eigenschaft BorderStyle	Richtet den Rahmen ein (einfach oder keiner).	B-6
Eigenschaft Enabled	Legt fest, ob das Control auf vom Anwender erzeugte Ereignisse reagiert.	B-19
Eigenschaft FalseCaption	Gibt den Text an, der im Control angezeigt wird, wenn die Eigenschaft "Value" auf "False" gesetzt ist (bzw. "0" oder "aus").	B-21
Eigenschaft FalseColor	Gibt die Farbe des Control an, wenn die Eigenschaft "Value" auf "False" gesetzt ist (bzw. "0" oder "aus").	B-21
Eigenschaft FalsePicture	Gibt das graphische Element an, das vom Control angezeigt wird, wenn der Wert der Eigenschaft Value "False" (gleich 0 oder "Off") ist.	B-22
Eigenschaft Font	Gibt ein Objekt "Font" für die Hauptschriftart des Control aus.	B-22
Eigenschaft ForeColor	Stellt die Vordergrundfarbe ein, in der Text des Control angezeigt wird.	B-23
Eigenschaft Locked	Versetzt das Control in einen schreibgeschützten Zustand. Standardmäßig ist das Control nicht verriegelt, so daß Sie Zahlen eingeben können.	B-26
Eigenschaft PushButton	Legt den Betriebszustand des Control fest: Ist Push-Button = True oder 1, wird die Eigenschaft Value invertiert, solange das Button Control gedrückt ist (Ereignis MouseDown).	B-32
Eigenschaft StretchMode	Gibt den Stretch-Modus des graphischen Elements des Control an.	B-38
Eigenschaft Style	Stellt die Darstellungsart (standard oder graphisch) des Control ein.	B-39
Eigenschaft TrueCaption	Gibt den Text an, der im Control angezeigt wird, wenn die Eigenschaft "Value" auf "True" gesetzt ist (bzw. "1" oder "ein").	B-40
Eigenschaft TrueColor	Gibt die Farbe des Control an, wenn die Eigenschaft "Value" auf "True" gesetzt ist (bzw. "1" oder "ein").	B-41
Eigenschaft TruePicture	Gibt das graphische Element an, das vom Control angezeigt wird, wenn der Wert der Eigenschaft Value "True" (gleich 1 oder "On") ist.	B-42
Eigenschaft Value	Enthält den Wert, der mit der Control Engine verknüpft ist.	B-43

6.4 Ereignisse des Button Control

Das Control reagiert auf die in Tabelle 6-2 aufgeführten Ereignisse.

Tabelle 6-2 Ereignisse des Button Control

Ereignis	Beschreibung	Seite
Ereignis Change	Tritt auf, wenn sich der Wert der Eigenschaft Value ändert.	C-1
Ereignis Click	Tritt auf, wenn sich der Mauszeiger auf dem Control befindet und eine Maustaste gedrückt und wieder losgelassen wird.	C-1
Ereignis Error	Tritt auf, wenn eine Eigenschaft auf einen unzulässigen Wert gesetzt ist.	C-2
Ereignis KeyDown	Tritt auf, wenn Sie eine Taste drücken, während ein Control aktiviert bzw. im Fokus ist.	C-3
Ereignis KeyPress	Tritt auf, wenn eine ANSI-Taste gedrückt und wieder losgelassen wird, während das Control aktiviert bzw. im Fokus ist.	C-5
Ereignis KeyUp	Tritt auf, wenn eine Taste gedrückt und wieder losgelassen wird, während das Control aktiviert bzw. im Fokus ist.	C-5
Ereignis MouseDown	Tritt auf, wenn sich der Mauszeiger auf dem Control befindet und die Maustaste gedrückt wird.	C-6
Ereignis MouseMove	Tritt auf, wenn Sie den Mauszeiger über ein Control bewegen.	C-7
Ereignis MouseUp	Tritt auf, wenn sich der Mauszeiger auf dem Control befindet und die Maustaste losgelassen wird.	C-8

Hinweis

Wenn Sie das Edit Control mit den tatsächlichen Prozeßdaten verbinden möchten, müssen Sie eine Verbindung über das Data Control herstellen.

6.5 Arbeiten mit dem Eigenschaftsdialog des Edit Control

Mit dem Edit Control können Sie Prozeßdaten in numerischem Format anzeigen und diese Daten ändern. Sie ordnen die numerische Anzeige Ihrem Prozeß zu, indem Sie der Anzeige eine Variable (den Prozeßwert) zuweisen. Sie können einen neuen Wert in die Anzeige eintragen. Die Anzeige wird automatisch aktualisiert, wenn sich die zugeordnete Variable im Prozeß ändert. Das Edit Control bietet Zugriff auf die Adressen im Speicher der Control Engine. Wenn Sie im Control einen neuen Wert eingeben, werden die Daten in der Control Engine geändert.

Hinweis

In SIMATIC Computing können Sie nicht in Zeiten schreiben.

Definieren, wie die Daten angezeigt werden (im Register "General")

Im Register "General" können Sie folgende Eigenschaften definieren und dadurch angeben, wie die Daten angezeigt werden:

- Die Eigenschaft "Alignment" definiert, wie der Wert im Edit Control angezeigt wird: linksbündig, zentriert oder rechtsbündig.
- Die Eigenschaft "Data Format" definiert die Speicherart, die für umgewandelte Werte verwendet wird. Wenn Sie einen Datentyp zum Anzeigen eines zu großen Werts verwenden, wird der Wert abgeschnitten.

Der in diesem Feld angegebene Datentyp muß mit dem im Feld "Assigned Variable" im Dialogfeld "Eigenschaften" für das S7Data Control eingestellten Datentyp übereinstimmen (siehe Bild 6-8). Tabelle 6-3 zeigt die Größen der Datentypen für das Edit Control.

- Die Eigenschaft "Precision" definiert die Nachkommastellen für Realzahlen (Gleitpunktzahlen). Sie geben an, wieviele Ziffern rechts vom Dezimalkomma angezeigt werden sollen. (Voreingestellt sind drei Nachkommastellen.) Dieses Feld ist nur für Werte vom Datentyp Real aktiviert.
- Die Eigenschaft "Appearance" definiert das Aussehen des Control. Wenn Sie für diese Eigenschaft "3D" einstellen, wird das Control in dreidimensionalem Format angezeigt. (Sie müssen dann auch für die Eigenschaft "Border Style" die Option "Fixed Single" einstellen, damit das dreidimensionale Format aktiviert werden kann.) Als zweite Möglichkeit können Sie "Flat" einstellen. Es wird dann ein zweidimensionaler rechteckiger Rahmen um das Control angezeigt.
- Die Eigenschaft "Border Style" definiert, ob ein Rahmen angezeigt wird. Wenn Sie für diese Eigenschaft den Wert "Fixed Single" einstellen, wird das Control mit rechteckigem Rahmen angezeigt. Wenn Sie für die Eigenschaft "None" einstellen, wird kein Rahmen angezeigt.
- Die Eigenschaft "WriteMode" legt die Reaktion des Control fest, wenn der Anwender einen neuen Wert eingibt. Ist die Eigenschaft WriteMode auf Automatic (0) gesetzt, wird der Wert (sofern er gültig ist) automatisch in die Eigenschaft Value (und in die Control Engine) geschrieben. Ist die Eigenschaft WriteMode auf Manual (1) gesetzt, dann wird der eingegebene Wert erst dann in die Eigenschaft Value geschrieben, wenn Ihr Programmcode die Methode "Write" für das Control aufruft.

Mit Hilfe der Kontrollkästchen im Register "General" können Sie andere Funktionen für das Control aktivieren:

- Das Kontrollkästchen "Enabled" legt fest, ob das Control auf Ereignisse reagiert. Es erzeugt im deaktivierten Zustand keine Ereignisse. Als Voreinstellung ist dieses Kontrollkästchen aktiviert (durch ein Häkchen gekennzeichnet).
- "Locked": Wenn Sie dieses Kontrollkästchen aktivieren, ist die Anzeige des Control schreibgeschützt. Sie können sich den Wert aus dem Speicher der Control Engine anzeigen lassen, doch Sie können die Werte im Control nicht ändern. Als Voreinstellung ist dieses Kontrollkästchen deaktiviert (nicht durch ein Häkchen gekennzeichnet).
- "Zero Pad": Wenn Sie dieses Kontrollkästchen aktivieren, füllt das Edit Control den Datentyp links vom Wert mit Nullen (0) auf. Als Voreinstellung ist dieses Kontrollkästchen deaktiviert (nicht durch ein Häkchen gekennzeichnet).

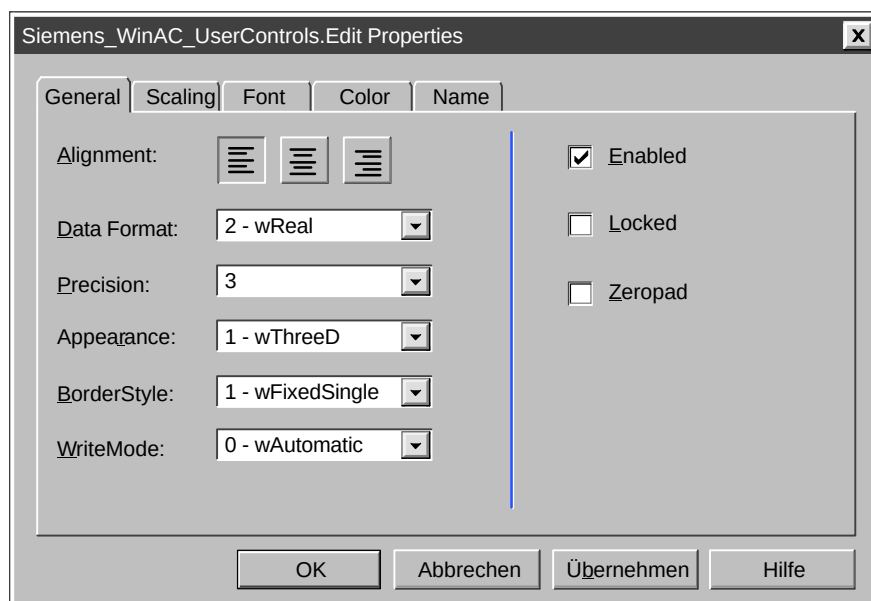


Bild 6-8 Eigenschaften des Edit Control (Register "General")

Tabelle 6-3 Größe der Datentypen für das Edit Control

Datentyp	Einstellung	Größe	Beschreibung
Boolean	0	1 Bit	Wert: ein Bit
Byte	1	1 Byte	Wert: vorzeichenlos, ein Byte
Word	2	1 Byte	Wert: vorzeichenlos, zwei Bytes
Integer	3	2 Bytes	Wert: mit Vorzeichen, zwei Bytes, ganze Zahl
Double Word	4	4 Bytes	Wert: vorzeichenlos, vier Bytes (Voreinstellung)
Double Integer	5	4 Bytes	Wert: mit Vorzeichen, vier Bytes, ganze Zahl
Real	6	4 Bytes	Wert: mit Vorzeichen, vier Bytes, Realzahl (Gleitpunktzahl)
Timer	7	2 Bytes	Wert: vorzeichenlos, zwei Bytes
Counter	8	2 Bytes	Wert: vorzeichenlos, zwei Bytes

Register "Scaling"

Im Register "Scaling" im Dialogfeld "Eigenschaften" (siehe Bild 6-9) können Sie eine Skalierung definieren, um den gespeicherten Wert anzuzeigen. Dieser Skalierungsfaktor wird sowohl beim Lesen von Werten aus der Control Engine als auch beim Schreiben von Werten in die Control Engine eingesetzt. Sie können zwischen drei Optionen zum Skalieren wählen:

- Keine Skalierung der Daten (Voreinstellung) (0-wNoScaling)
- Skalierung nach Formel (1-wByFormula)
- Skalierung durch Bereichsumwandlung (2-wByRange)

Keine Skalierung der Daten: Wenn Sie die Voreinstellung wählen, zeigt "Display Value" ein Maximum von 100 und ein Minimum von 0 an.

Skalierung durch Formel: Wenn Sie mit Hilfe einer Formel skalieren möchten, geben Sie folgende Informationen an:

- Factor stellt die Änderung in Prozent (Skalierungsfaktor) zwischen dem Wert in der Control Engine und dem Wert im Edit Control dar.
- Offset stellt einen festen Wert dar, der zum Skalierungsergebnis addiert werden muß, bevor der Wert angezeigt werden kann.

Das Edit Control setzt folgende Formel ein, um den skalierten Wert zu berechnen:

$$(\text{Value} \times \text{Factor}) + \text{Offset} = \text{Display Value}$$

Erklärung:

Value = in der Control Engine gespeicherter Wert

Factor = Skalierungsfaktor

Offset = Offset-Faktor

Display Value = im Edit Control angezeigter Wert

Wenn das Edit Control Daten in die Control Engine schreibt, wird die umgekehrte Formel verwendet, um den Wert zu skalieren.

Skalierung durch Bereichsumwandlung: Wenn Sie durch Bereichsumwandlung skalieren möchten, geben Sie die unteren (RawMax) und oberen (RawMin) Grenzwerte des Quellbereichs an (für den Wert in der Control Engine) und auch für den Zielbereich (für den Wert der vom Edit Control angezeigt werden soll). Das Edit Control wandelt den Wert von einem Bereich in den entsprechenden Wert im anderen Bereich um.

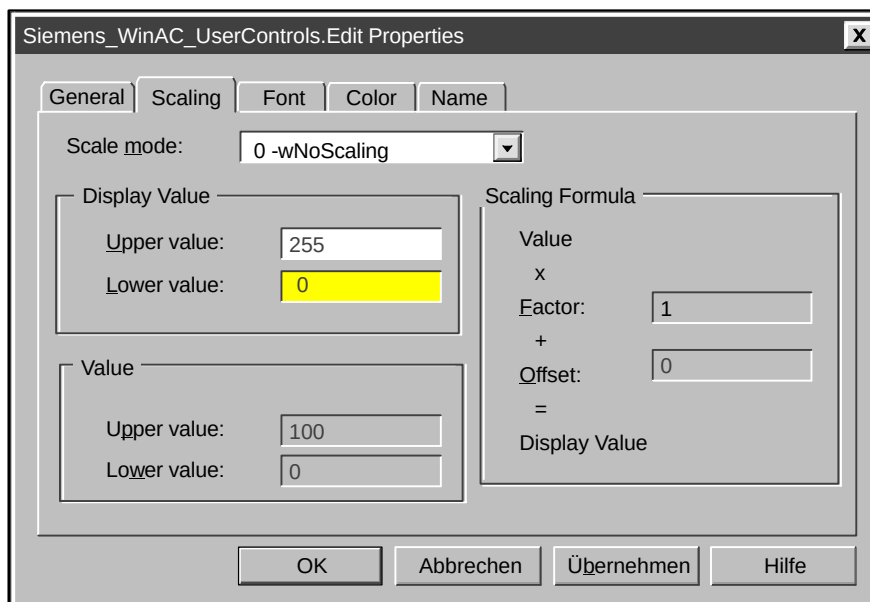


Bild 6-9 Eigenschaften des Edit Control (Register "Scaling")

Diese Bereiche definieren nur die Beziehung zwischen den Daten in der Control Engine und den Daten im Edit Control. Liegt der Wert oberhalb oder unterhalb der Bereichsgrenzen, die für die Umwandlung eingegeben wurden, dann wird der skalierte Wert während der Umwandlung anhand einer Formel extrapoliert. Die oberen und unteren Grenzwerte sind keine Mindest- und Höchstwerte ("minimum and maximum values") für die Daten. Die Skalierungsfaktoren werden nicht auf Grenzwerte geprüft.

Definieren der Schrift des Texts (im Register "Font")

Im Register "Font" im Dialogfeld "Eigenschaften" (siehe Bild 6-10) können Sie die Schriftart und die Schriftgröße für den Text auf dem Edit Control definieren:

- Die Eigenschaft "Font" wählt aus einer Liste von Standard-Schriftarten eine für den Text aus.
- Wählen Sie unter "Size" die Größe in Punkt aus oder geben Sie eine bestimmte Größe in Punkt für den Text ein.
- Wählen Sie unter "Effects" noch andere typographische Optionen (fett, kursiv, unterstrichen, durchgestrichen) für den Text aus.

Das Feld "Sample Text" zeigt ein Beispiel für die in der Eigenschaft Font ausgewählte Schriftart an.

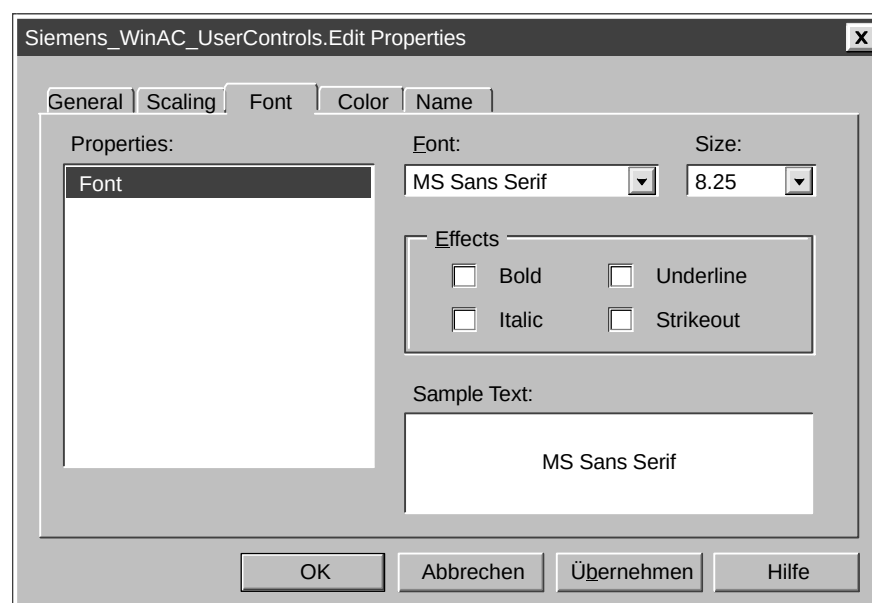


Bild 6-10 Eigenschaften des Edit Control (Register "Font")

Definieren der Farbe für das Edit Control (im Register "Color")

Im Register "Color" im Dialogfeld "Eigenschaften" (siehe Bild 6-11) können Sie die Farben für die beiden Zustände und für den Text des Edit Control definieren: Sie wählen die Eigenschaft ("BackColor" oder "ForeColor") aus und wählen dann die Farbe für die jeweilige Eigenschaft aus der Farbpalette aus. Sie können aus den Standardfarben auswählen, doch Sie können auch Ihre eigenen Farben erstellen.

Hinweis

Die Eigenschaften BackColor und ForeColor können nur geändert werden, wenn Sie im Register "General" Style: Standard ausgewählt haben. Die Eigenschaft ForeColor (Farbe des Texts) können Sie sowohl für Style: Standard und für Style: Graphical ändern.

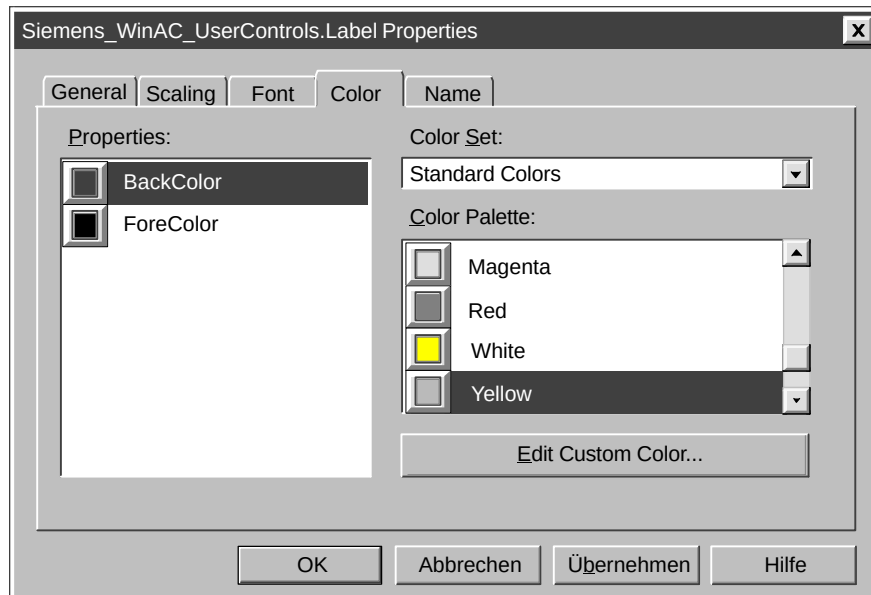


Bild 6-11 Eigenschaften des Edit Control (Register "Color")

Register "Name"

Im Register "Name" des Dialogfelds "Eigenschaften" (siehe Bild 6-12) können Sie den Controls, die Sie in den Container eingefügt haben, beschreibende Namen zuordnen. Dieses Register ist nur verfügbar, wenn Sie mit Controls im SIMATIC Computing Container arbeiten.

Sie geben den neuen Namen in das Feld "Control Name" ein und bestätigen mit "Übernehmen" oder "OK". Der neue Name wird nun in der Liste "Select Control" in der Funktionsleiste aufgeführt. Durch Doppelklick auf das gewünschte Control rufen Sie die Eigenschaften des Control im Dialogfeld "Eigenschaften" auf.

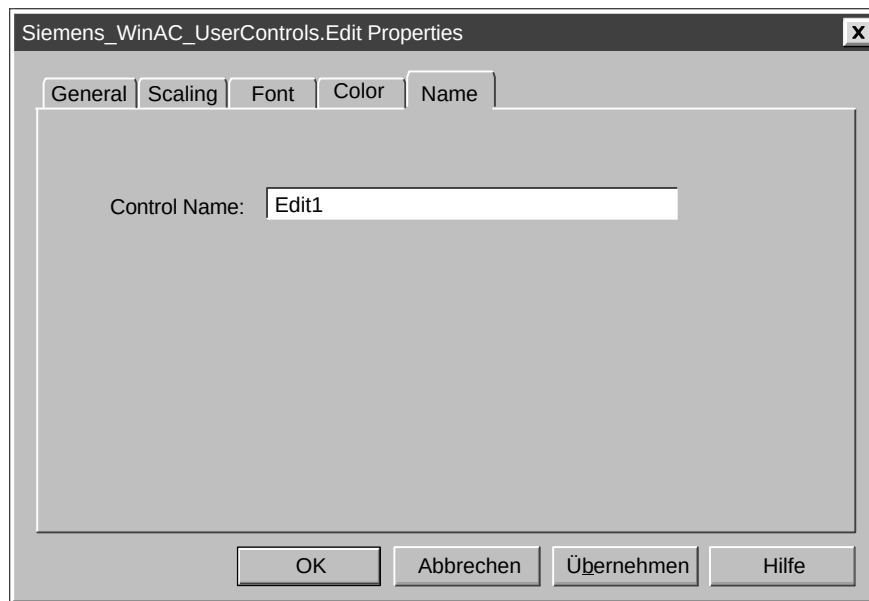


Bild 6-12 Eigenschaften des Edit Control (Register "Name")

6.6 Eigenschaften und Methoden des Edit Control

Mit den in Tabelle 6-4 aufgeführten Eigenschaften und Methoden können Sie das Edit Control bearbeiten.

Tabelle 6-4 Eigenschaften und Methoden des Edit Control

Eigenschaft / Methode	Beschreibung	Seite
Methode AboutBox	Zeigt das Dialogfeld "Info" des Control an.	B-1
Eigenschaft Alignment	Gibt die Ausrichtung der Zahl im Control an.	B-2
Eigenschaft Appearance	Gibt an, ob das Control dreidimensional oder "flach" dargestellt wird.	B-3
Eigenschaft BackColor	Stellt die Hintergrundfarbe ein.	B-5
Eigenschaft BorderStyle	Richtet den Rahmen ein (einfach oder keiner).	B-6
Eigenschaft DataFormat	Definiert die Speicherart, die für umgewandelte Werte verwendet wird.	B-11
Eigenschaft DisplayValue	Gibt den skalierten Wert für das Control aus.	B-18
Eigenschaft Enabled	Gibt an, ob das Control auf Änderungen in der Eigenschaft Value reagiert und Ereignisse auslöst.	B-19
Eigenschaft Factor	Gibt den Skalierfaktor an, der benötigt wird, wenn Sie mit Hilfe einer Formel skalieren möchten (in Zusammenhang mit der Eigenschaft ScaleMode).	B-20
Eigenschaft Font	Gibt ein Objekt "Font" für die Hauptschriftart des Control aus.	B-22
Eigenschaft ForeColor	Stellt die Vordergrundfarbe ein, in der Text und Graphiken angezeigt werden.	B-23
Eigenschaft Locked	Versetzt das Control in einen schreibgeschützten Zustand. Standardmäßig ist das Control nicht verriegelt, so daß Sie Zahlen eingeben können.	B-26
Eigenschaft Max	Gibt den höchsten skalierten Wert des Control an, wenn die Eigenschaft ScaleMode auf wByRange oder wScaleNone gesetzt ist.	B-26
Eigenschaft Min	Gibt den niedrigsten skalierten Wert des Control an, wenn die Eigenschaft ScaleMode auf wByRange oder wScaleNone gesetzt ist.	B-26
Eigenschaft Offset	Gibt den Offset an, der benötigt wird, wenn Sie mit Hilfe einer Formel skalieren möchten (in Zusammenhang mit der Eigenschaft ScaleMode).	B-27
Eigenschaft Precision	Gibt die Genauigkeit der Realzahl an.	B-30
Eigenschaft RawMax	Definiert den oberen Wert des Quellbereichs zum Skalieren eines Werts. Die Eigenschaft ScaleMode muß auf "wByRange" gesetzt sein.	B-32
Eigenschaft RawMin	Definiert den unteren Wert des Quellbereichs zum Skalieren eines Werts. Die Eigenschaft ScaleMode muß auf "wByRange" gesetzt sein.	B-32
Eigenschaft ScaleMode	Gibt an, auf welche Art Werte skaliert werden sollen.	B-34

Tabelle 6-4 Eigenschaften und Methoden des Edit Control, Fortsetzung

Eigenschaft / Methode	Beschreibung	Seite
Eigenschaft Value	Enthält den Wert, der mit der Control Engine verknüpft ist.	B-43
Eigenschaft WriteMode	Gibt an, ob neue Werte automatisch oder manuell geschrieben werden sollen.	B-43
Methode WriteNow	Schreibt den Wert der Eigenschaft Value	B-44
Eigenschaft ZeroPad	Gibt an, ob die angezeigte Zahl bis zur Größe des Datentyps mit Nullen aufgefüllt wird (links vom Wert).	B-46

6.7 Ereignisse des Edit Control

Das Edit Control reagiert auf die in Tabelle 6-5 aufgeführten Ereignisse.

Tabelle 6-5 Ereignisse des Edit Control

Ereignis	Beschreibung	Seite
Ereignis Change	Tritt auf, wenn sich der Wert der Eigenschaft Value ändert.	C-1
Ereignis Click	Tritt auf, wenn sich der Mauszeiger auf dem Control befindet und eine Maustaste gedrückt und wieder losgelassen wird.	C-1
Ereignis DblClick	Tritt auf, wenn sich der Mauszeiger auf dem Control befindet und mit einer Maustaste doppelt geklickt wird.	C-2
Ereignis Error	Tritt auf, wenn eine Eigenschaft auf einen unzulässigen Wert gesetzt ist.	C-2
Ereignis KeyDown	Tritt auf, wenn Sie eine Taste drücken, während ein Control aktiviert bzw. im Fokus ist.	C-3
Ereignis KeyPress	Tritt auf, wenn eine ANSI-Taste gedrückt und wieder losgelassen wird, während das Control aktiviert bzw. im Fokus ist.	C-5
Ereignis KeyUp	Tritt auf, wenn eine Taste gedrückt und wieder losgelassen wird, während das Control aktiviert bzw. im Fokus ist.	C-5
Ereignis MouseDown	Tritt auf, wenn sich der Mauszeiger auf dem Control befindet und die Maustaste gedrückt wird.	C-6
Ereignis MouseMove	Tritt auf, wenn Sie den Mauszeiger über ein Control bewegen.	C-7
Ereignis MouseUp	Tritt auf, wenn sich der Mauszeiger auf dem Control befindet und die Maustaste losgelassen wird.	C-8

6.8 Fehlercodes des Edit Control

Tritt im Edit Control ein Fehler auf, erzeugt das Control ein Fehlerereignis. Ihr Programm kann das Ereignis Error aufnehmen und auf bestimmte Situationen reagieren. Tabelle 6-6 führt die Fehlercodes für das Edit Control auf.

Tabelle 6-6 Fehlercodes des Edit Control

Fehlercode	Beschreibung
C0040002	Die Skalierung kann wegen eines Fehlers in der verwendeten Formel nicht fortgesetzt werden. Dieser Fehler tritt nur auf, wenn Sie für das Edit Control mit Skalierung durch Bereichsumwandlung arbeiten. In diesem Fall kann es sein, daß Sie für den Bereich eines Rohwerts (RawMin, RawMax) die Länge Null (<i>min</i> gleich <i>max</i>) angegeben haben. Dies führt zur Division durch Null, wodurch die Skalierung nicht möglich ist. Zum Beheben des Fehlers geben Sie einen neuen Rohwert an, wobei RawMin ungleich RawMax sein muß.
C0040003	Der für die Eigenschaft Value eingestellte Wert ist unzulässig. Der Wert aus der Control Engine bzw. aus einem Skript, das auf die Eigenschaft Value zugreift, kann nicht ausgewertet werden. Zum Beheben des Fehlers prüfen und korrigieren Sie die Werte, die Sie in das Control geschrieben haben.
C0040004	Der für die Eigenschaft Text eingestellte Wert ist unzulässig. Hier handelt es sich um einen Fehler, der häufig auftritt, und zwar dann, wenn der Anwender einen ungültigen Wert in das Control eingibt. Normalerweise ist der Fehler darauf zurückzuführen, daß der eingegebene Text unzulässige Zeichen enthält. Die zulässigen Zeichen richten sich nach dem Datentyp. Zum Beheben des Fehlers geben Sie einen zulässigen Wert ein.
C0040005	Die anderen OLE-Komponenten konnten nicht gefunden werden. Es ist ein Fehler in der Installation von SIMATIC Computing oder in Windows aufgetreten. Das Control kann nicht auf andere erforderliche Teile zugreifen, die zur fehlerfreien Funktionsweise der Software benötigt werden. Zum Beheben des Fehlers prüfen Sie die Installation.
C0040006	Die Microsoft Standard Controls konnten nicht erstellt werden. Es ist ein Fehler in der Installation von SIMATIC Computing oder in Windows aufgetreten. Das Control kann nicht auf andere erforderliche Teile zugreifen, die zur fehlerfreien Funktionsweise der Software benötigt werden. Zum Beheben des Fehlers prüfen Sie die Installation.
C0040010	Die Grenzwertprüfung kann nicht fortgesetzt werden, weil RawMin größer ist als der RawMax. Dieser Fehler kann nur auftreten, wenn Sie im Edit Control mit Grenzwertprüfung arbeiten (Prüfen der oberen und unteren Grenzwerte). In diesem Fall kann es sein, daß Sie einen unteren Grenzwert (RawMin) eingegeben haben, der größer ist als der obere Grenzwert (RawMax). Zum Beheben des Fehlers geben Sie einen gültigen Bereich für die Grenzwertprüfung an. Der untere Grenzwert muß kleiner sein als der obere Grenzwert.

6.9 Arbeiten mit dem Eigenschaftsdialog des Label Control

Mit dem Label Control können Sie eine Konstante als Zeichenfolge anzeigen. Sie können die Eigenschaft Caption des Label Control auch mit einem beliebigen Prozeßwert verbinden. Der Prozeßwert wird in eine Zeichenfolge umgewandelt und angezeigt. Das Label Control kann nicht als Eingang eingesetzt werden.

Definieren des Labels und Freigeben des Control (im Register "General")

Im Register "General" im Dialogfeld "Eigenschaften" (siehe Bild 6-13) können Sie die Darstellung des Label Control definieren:

- Die Eigenschaft "Alignment" definiert, wie der Wert im Label Control angezeigt wird: linksbündig, zentriert oder rechtsbündig.
- Die Eigenschaft "Caption" gibt den Text an, der vom Control angezeigt werden soll. Ist die Eigenschaft Caption einem Prozeßwert zugeordnet, wird der Prozeßwert angezeigt.
- Die Eigenschaft "Style" stellt die Darstellungsart (standard oder graphisch) des Control ein.
- Die Eigenschaft "Appearance" definiert das Aussehen des Control. Wenn Sie für diese Eigenschaft "3D" einstellen, wird das Control in dreidimensionalem Format angezeigt. (Sie müssen dann auch für die Eigenschaft "Border Style" die Option "Fixed Single" einstellen, damit das dreidimensionale Format aktiviert werden kann.) Als zweite Möglichkeit können Sie "Flat" einstellen. Es wird dann ein zweidimensionaler rechteckiger Rahmen um das Control angezeigt.
- Die Eigenschaft "Border Style" definiert, ob ein Rahmen angezeigt wird. Wenn Sie für diese Eigenschaft den Wert "Fixed Single" einstellen, wird das Control mit rechteckigem Rahmen angezeigt. Wenn Sie für die Eigenschaft "None" einstellen, wird kein Rahmen angezeigt.
- Die Eigenschaft "StretchMode" gibt den Stretch-Modus des graphischen Elements des Control an.
- Das Kontrollkästchen "Enabled" legt fest, ob das Label Control auf Ereignisse reagiert. Es erzeugt im deaktivierten Zustand keine Ereignisse. Als Voreinstellung ist dieses Kontrollkästchen aktiviert (durch ein Häkchen gekennzeichnet).

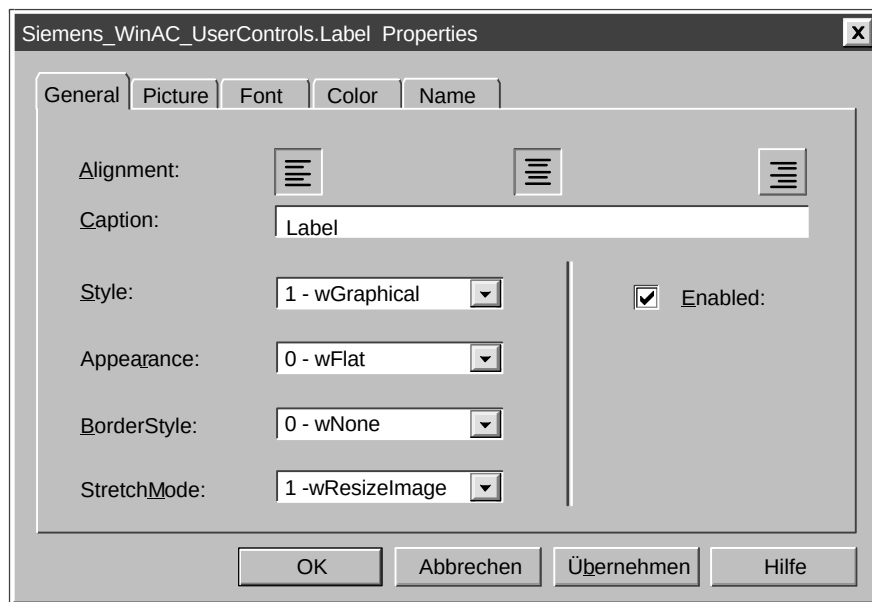


Bild 6-13 Eigenschaften des Label Control (Register "General")

Definieren der Abbildung für das Label Control (im Register "Picture")

Im Register "Picture" im Dialogfeld "Eigenschaften" (siehe Bild 6-14) können Sie ein Bild für die beiden Zustände des Label Control auswählen: Sie markieren "Picture" und wählen dann über die Schaltfläche "Browse" das Bild für den jeweiligen Zustand aus. Achten Sie darauf, daß Sie die Eigenschaft Picture nur verwenden können, wenn Sie im Register "General" Style: Graphical gewählt haben. Vordefinierte Bitmaps befinden sich im Verzeichnis WinACWinCP\bitmaps.

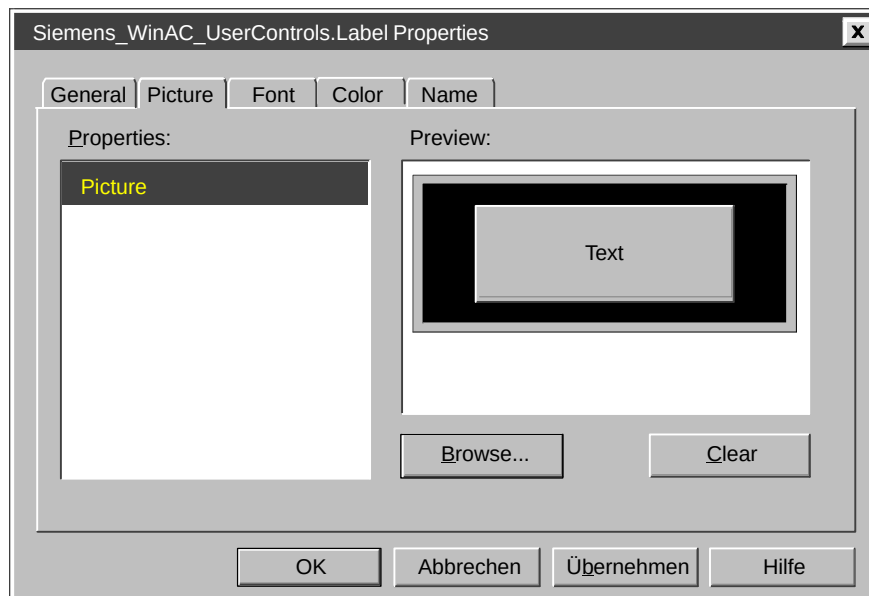


Bild 6-14 Eigenschaften des Label Control (Register "Picture")

Definieren der Schriftart für das Label Control (im Register "Font")

Im Register "Font" im Dialogfeld "Eigenschaften" (siehe Bild 6-15) können Sie die Schriftart und die Schriftgröße der beiden Labels des Label Control definieren:

- "Font": Wählen Sie hier aus einer Liste von Standard-Schriftarten eine für das Label aus.
- "Size": Wählen Sie hier die Größe in Punkt für das Label aus oder geben Sie eine bestimmte Größe in Punkt für das Label ein.
- "Effects": Wählen Sie hier noch andere typographische Optionen (fett, kursiv, unterstrichen, durchgestrichen) für das Label aus.

Das Feld "Sample Text" zeigt ein Beispiel für die in der Eigenschaft Font ausgewählte Schriftart an.

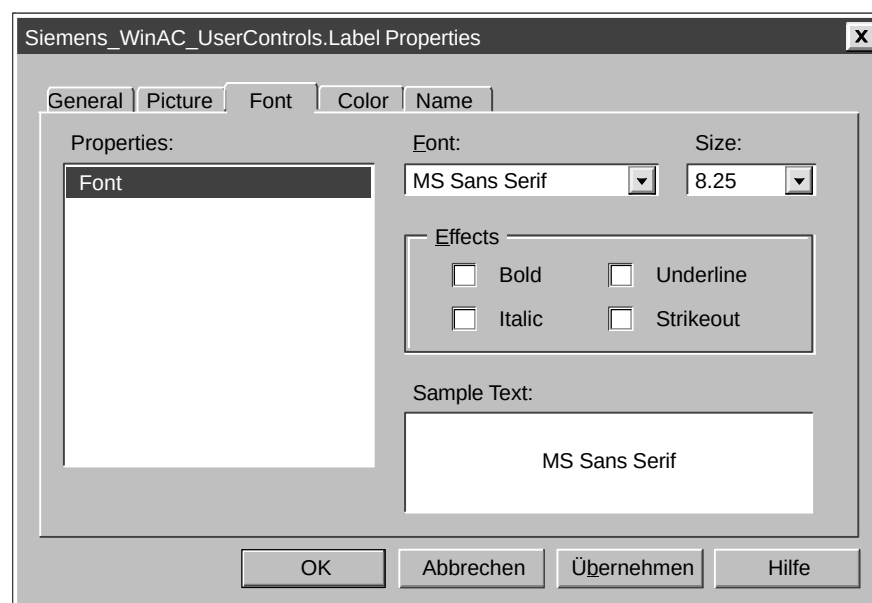


Bild 6-15 Eigenschaften des Label Control (Register "Font")

Definieren der Farbe für das Label Control (im Register "Color")

Im Register "Color" im Dialogfeld "Eigenschaften" (siehe Bild 6-16) können Sie die Farben für den Hintergrund (BackColor) und für den Text (ForeColor) des Label Control definieren. Sie wählen die Eigenschaft ("BackColor oder "ForeColor") aus und wählen dann die Farbe für die jeweilige Eigenschaft aus der Farbpalette aus. Sie können aus den Standardfarben auswählen, doch Sie können auch Ihre eigenen Farben erstellen.

Hinweis

ForeColor (Farbe des Texts) kann für Style: Standard und für Style: Graphical geändert werden, doch die Farbe für den Hintergrund (BackColor) kann je nach Stretch-Modus vom Bitmap des Label Control verdeckt sein.

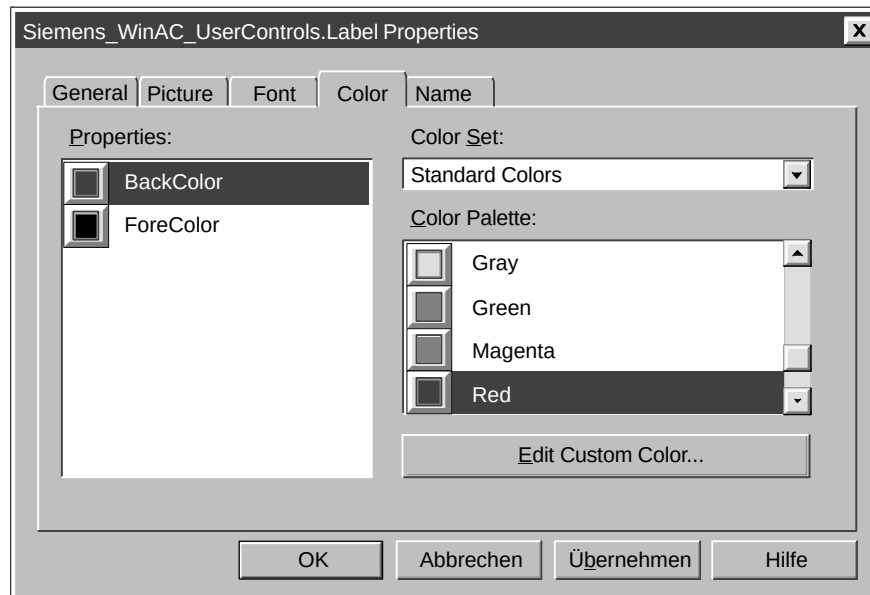


Bild 6-16 Eigenschaften des Label Control (Register "Color")

Register "Name"

Im Register "Name" des Dialogfelds "Eigenschaften" (siehe Bild 6-17) können Sie den Controls, die Sie in den Container eingefügt haben, beschreibende Namen zuordnen. Dieses Register ist nur verfügbar, wenn Sie mit Controls im SIMATIC Computing Container arbeiten.

Sie geben den neuen Namen in das Feld "Control Name" ein und bestätigen mit "Übernehmen" oder "OK". Der neue Name wird nun in der Liste "Select Control" in der Funktionsleiste aufgeführt. Durch Doppelklick auf das gewünschte Control rufen Sie die Eigenschaften des Control im Dialogfeld "Eigenschaften" auf.

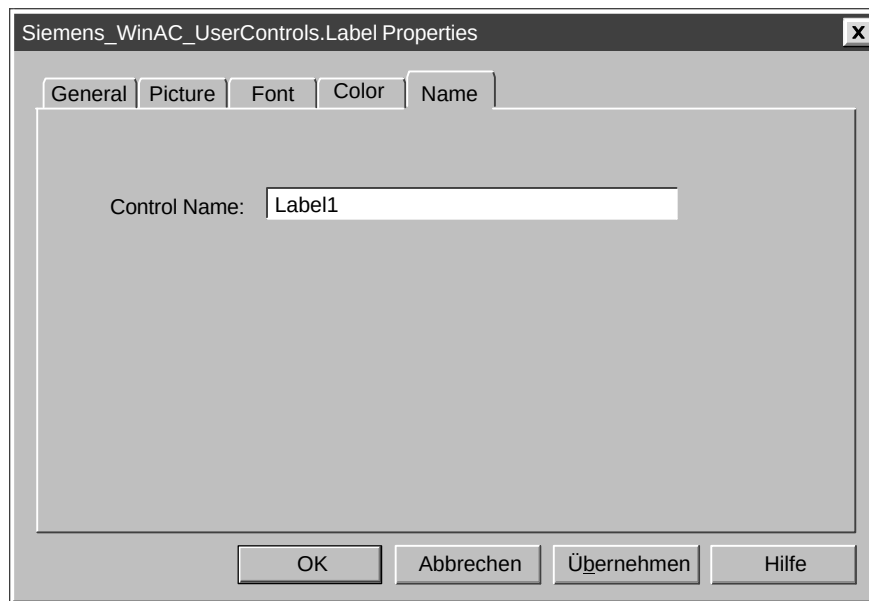


Bild 6-17 Eigenschaften des Label Control (Register "Name")

6.10 Eigenschaften und Methoden des Label Control

Mit den in Tabelle 6-7 aufgeführten Eigenschaften und Methoden können Sie das Label Control bearbeiten.

Tabelle 6-7 Eigenschaften und Methoden des Label Control

Eigenschaft / Methode	Beschreibung	Seite
Methode AboutBox	Zeigt das Dialogfeld "Info" des Control an.	B-1
Eigenschaft Alignment	Legt die Ausrichtung des Texts fest.	B-2
Eigenschaft Appearance	Legt fest, ob das Control mit 3D-Effekt angezeigt wird.	B-3
Eigenschaft BackColor	Legt die Hintergrundfarbe des Control fest.	B-5
Eigenschaft BorderStyle	Richtet den Rahmen ein (einfach oder keiner).	B-6
Eigenschaft Caption	Gibt den Text an, der vom Control angezeigt werden soll.	B-7
Eigenschaft Enabled	Legt fest, ob das Control auf vom Anwender erzeugte Ereignisse reagiert.	B-19
Eigenschaft Font	Gibt ein Objekt "Font" für die Hauptschriftart des Control aus.	B-22
Eigenschaft ForeColor	Legt die Farbe für den Text des Control fest.	B-23
Eigenschaft Picture	Legt das Bild (die Graphik) für das Control fest.	B-28
Eigenschaft StretchMode	Gibt den Stretch-Modus des graphischen Elements des Control an.	B-38
Eigenschaft Style	Stellt die Darstellungsart (standard oder graphisch) des Control ein.	B-39

6.11 Ereignisse des Label Control

Das Label Control reagiert auf die in Tabelle 6-8 aufgeführten Ereignisse.

Tabelle 6-8 Ereignisse des Edit Control

Ereignis	Beschreibung	Seite
Ereignis Change	Tritt auf, wenn sich der Wert der Eigenschaft Value ändert.	C-1
Ereignis Click	Tritt auf, wenn sich der Mauszeiger auf dem Control befindet und eine Maustaste gedrückt und wieder losgelassen wird.	C-1
Ereignis DblClick	Tritt auf, wenn sich der Mauszeiger auf dem Control befindet und mit einer Maustaste doppelt geklickt wird.	C-2
Ereignis Error	Tritt auf, wenn eine Eigenschaft auf einen unzulässigen Wert gesetzt ist.	C-2
Ereignis MouseDown	Tritt auf, wenn sich der Mauszeiger auf dem Control befindet und die Maustaste gedrückt wird.	C-6

Tabelle 6-8 Ereignisse des Edit Control, Fortsetzung

Ereignis	Beschreibung	Seite
Ereignis MouseMove	Tritt auf, wenn Sie den Mauszeiger über ein Control bewegen.	C-7
Ereignis MouseUp	Tritt auf, wenn sich der Mauszeiger auf dem Control befindet und die Maustaste losgelassen wird.	C-8

6.12 Arbeiten mit dem Eigenschaftsdialog des Slider Control

Mit dem Slider Control können Sie Prozeßdaten visuell anzeigen (als Bildlaufanzeige) und diese Daten ändern. Sie ordnen das Bildlauffeld Ihrem Prozeß zu, indem Sie dem Bildlauffeld eine Variable (den Prozeßwert) zuweisen. Sie können dann über die Bildlaufanzeige den Prozeßwert ändern. Die Bildlaufanzeige im Bildlauffeld ändert sich automatisch, wenn sich die zugeordnete Variable im Prozeß ändert.

Das Slider Control bietet Zugriff auf die Adressen im Speicher der Control Engine. Wenn Sie im Control einen neuen Wert eingeben, werden die Daten in der Control Engine geändert.

Hinweis

In SIMATIC Computing können Sie nicht in Zeiten schreiben.

Definieren, wie die Daten angezeigt werden (im Register "General")

Im Register "General" im Dialogfeld "Eigenschaften" (siehe Bild 6-18) können Sie die Darstellungsart der Daten, auf die das Slider Control zugreift, definieren.

Mit Hilfe der Felder im Register "General" können Sie folgende Eigenschaften definieren:

- Die Eigenschaft "Style" stellt die Darstellungsart (standard oder graphisch) des Control ein.
- Die Eigenschaft "Direction" stellt die Ausrichtung (waagerecht oder senkrecht) des Control ein (siehe Bild 6-19).
- Die Eigenschaft "StretchMode" gibt den Stretch-Modus des graphischen Elements des Control an.
- Die Eigenschaft "Ticks" gibt die Anzahl der Teilstriche zwischen dem Mindest- und dem Höchstwert an.
- Die Eigenschaften "SmallChange" und "LargeChange" geben an, um wieviel der vom Slider Control angezeigte Wert erhöht bzw. verringert wird, wenn Sie eine der Richtungstasten (SmallChange) oder eine der Tasten Seite-nach-oben oder Seite-nach-unten (LargeChange) drücken.
- Die Eigenschaften "KnobHeight" und "KnobWidth" geben die Höhe und Breite der vom Control angezeigten Anzeige an.

Mit Hilfe der Kontrollkästchen im Register "General" können Sie andere Funktionen für das Control aktivieren:

- Das Kontrollkästchen "Show Min. and Max Value" gibt an, ob die Mindest- und Höchstwerte angezeigt werden.
- Das Kontrollkästchen "Enabled" legt fest, ob das Control auf Ereignisse reagiert. Es erzeugt im deaktivierten Zustand keine Ereignisse. Als Voreinstellung ist dieses Kontrollkästchen aktiviert (durch ein Häkchen gekennzeichnet).
- Das Kontrollkästchen "Locked" legt fest, ob das Control schreibgeschützt ist. Im verriegelten Zustand können Sie keine Werte ändern.

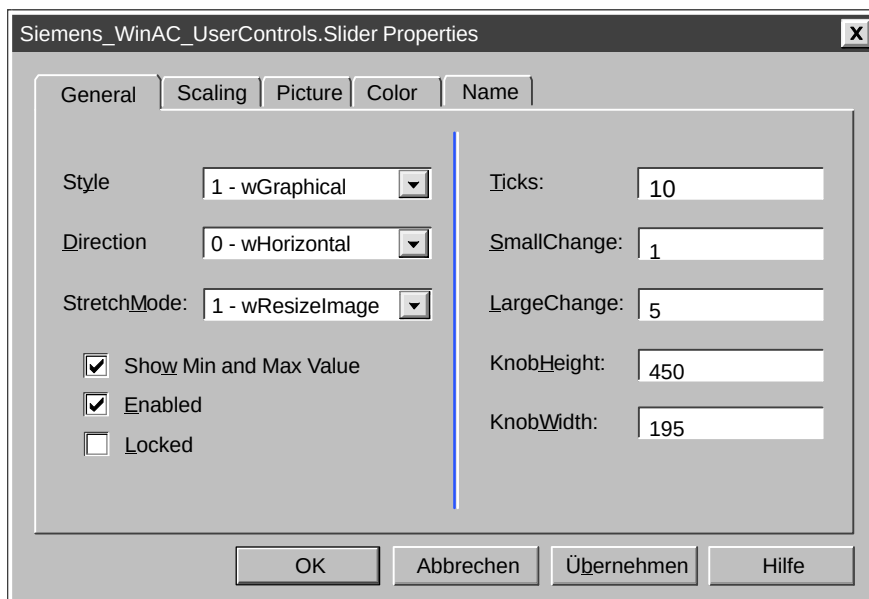


Bild 6-18 Eigenschaften des Slider Control (Register "General")

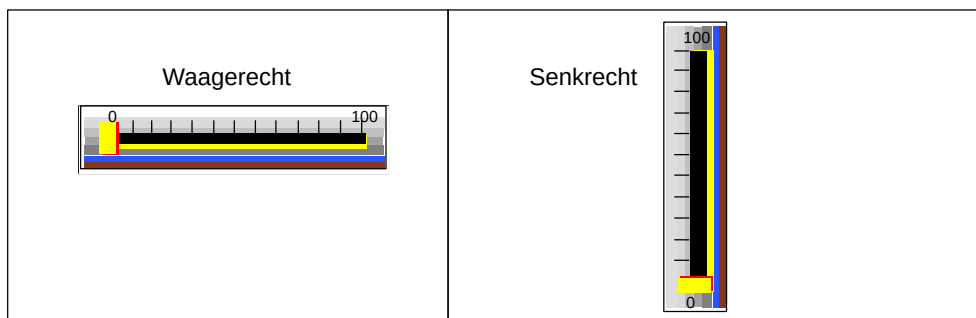


Bild 6-19 Ausrichtung des Slider Control

Register "Scaling"

Im Register "Scaling" im Dialogfeld "Eigenschaften" (siehe Bild 6-21) können Sie eine Skalierung definieren, um den gespeicherten Wert anzuzeigen. Dieser Skalierungsfaktor wird sowohl beim Lesen von Werten aus der Control Engine als auch beim Schreiben von Werten in die Control Engine eingesetzt. Sie können zwischen drei Optionen zum Skalieren wählen:

- Keine Skalierung der Daten (Voreinstellung) (0-wNoScaling)
- Skalierung nach Formel (1-wByFormula)
- Skalierung durch Bereichsumwandlung (2-wByRange)

Keine Skalierung der Daten: Wenn Sie die Voreinstellung wählen, zeigt "Display Value" ein Maximum von 100 und ein Minimum von 0 an.

Skalierung durch Formel: Wenn Sie mit Hilfe einer Formel skalieren möchten, geben Sie folgende Informationen an:

- Factor stellt die Änderung in Prozent (Skalierungsfaktor) zwischen dem Wert in der Control Engine und dem Wert im Slider Control dar.
- Offset stellt einen festen Wert dar, der zum Skalierungsergebnis addiert werden muß, bevor der Wert angezeigt werden kann.

Das Slider Control setzt die folgende Formel ein, um den skalierten Wert zu berechnen:

$$(\text{Value} \times \text{Factor}) + \text{Offset} = \text{Display Value}$$

Erklärung:

PLC value = in der Control Engine gespeicherter Wert

Scale = Skalierungsfaktor

Offset = Offset-Faktor

Display = vom Number Control angezeigter Wert

Wenn das Slider Control Daten in die Control Engine schreibt, wird die umgekehrte Formel verwendet, um den Wert zu skalieren.

Skalierung durch Bereichsumwandlung: Wenn Sie durch Bereichsumwandlung skalieren möchten, geben Sie die unteren (RawMax) und oberen (RawMin) Werte des Quellbereichs (Felder "Value") und des Zielbereichs (Felder "Display Value") an. Das Slider Control wandelt den Wert von einem Bereich in den entsprechenden Wert im anderen Bereich um.

Diese Bereiche definieren nur die Beziehung zwischen den Daten in der Control Engine und den Daten im Slider Control. Liegt der Wert oberhalb oder unterhalb der Bereichsgrenzen, die für die Umwandlung eingegeben wurden, dann wird der skalierte Wert während der Umwandlung anhand einer Formel extrapoliert. Die oberen und unteren Grenzwerte sind keine Mindest- und Höchstwerte ("minimum and maximum values") für die Daten. Die Skalierungsfaktoren werden nicht auf Grenzwerte geprüft.

Bild 6-20 zeigt die angezeigten Werte des Slider Control.

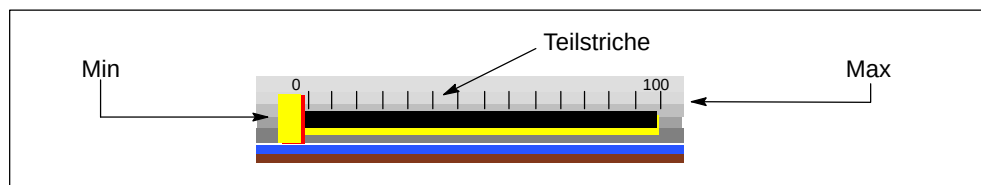


Bild 6-20 Elemente des Slider Control

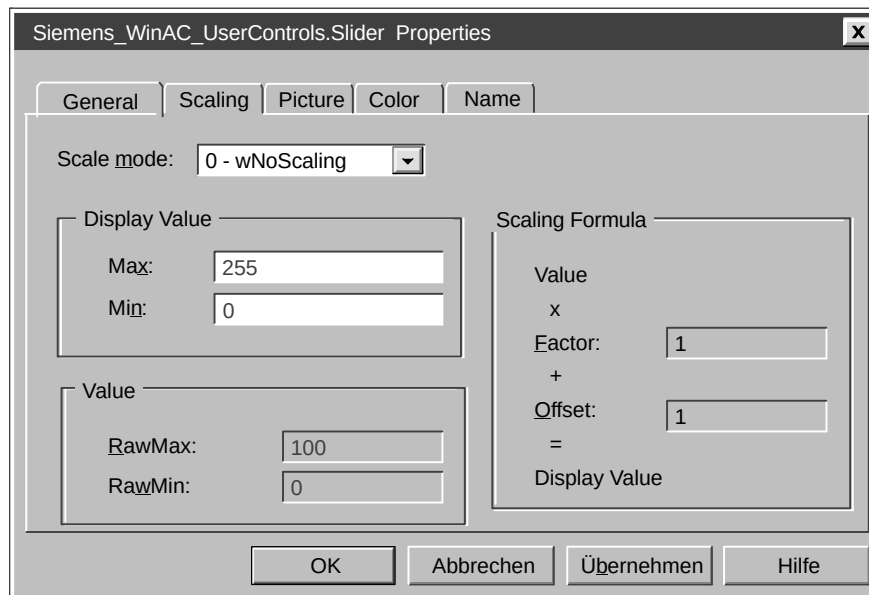


Bild 6-21 Eigenschaften des Slider Control (Register "Scaling")

Definieren der Abbildung für das Slider Control (im Register "Picture")

Im Register "Picture" im Dialogfeld "Eigenschaften" (siehe Bild6-14) können Sie die Bilder für das Slider Control auswählen: Sie wählen "KnobPicture" und anschließend die Schaltfläche "Browse", um das Bild (die Graphik) für die Anzeige auf dem Control auszuwählen. Wählen Sie dann "Picture" und anschließend "Browse", um das Bild (die Graphik) für das Control auszuwählen. Vordefinierte Bitmaps befinden sich im Verzeichnis WinAC\WinCP\bitmaps.

Hinweis

Die Eigenschaft "Picture" kann nur geändert werden, wenn Sie im Register "General" Style: Graphical gewählt haben. Die Eigenschaft "KnobPicture" können Sie für Style: Standard und für Style: Graphical ändern.

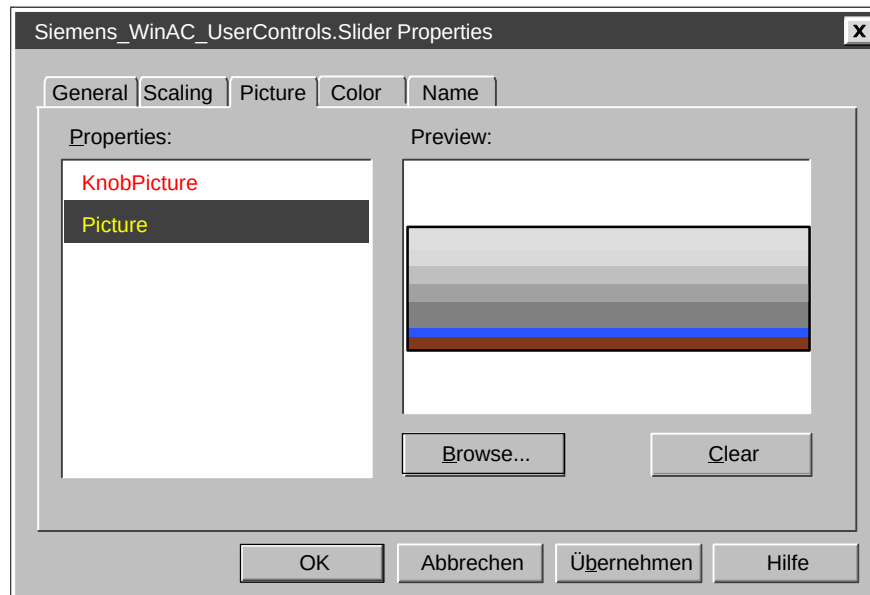


Bild 6-22 Eigenschaften des Slider Control (Register "Picture")

Definieren der Farbe für das Slider Control (im Register "Color")

Im Register "Color" im Dialogfeld "Eigenschaften" (siehe Bild 6-11) können Sie die beiden Farben (BackColor und ForeColor) für den Text des Slider Control definieren. Sie wählen die Eigenschaft ("BackColor oder "ForeColor") aus und wählen dann die Farbe für die jeweilige Eigenschaft aus der Farbpalette aus. Sie können aus den Standardfarben auswählen, doch Sie können auch Ihre eigenen Farben erstellen.

- Die Eigenschaft "BackColor" gibt die Hintergrundfarbe für das Control an.
- Die Eigenschaft "ForeColor" gibt die Farbe für Text und Graphiken im Objekt an.

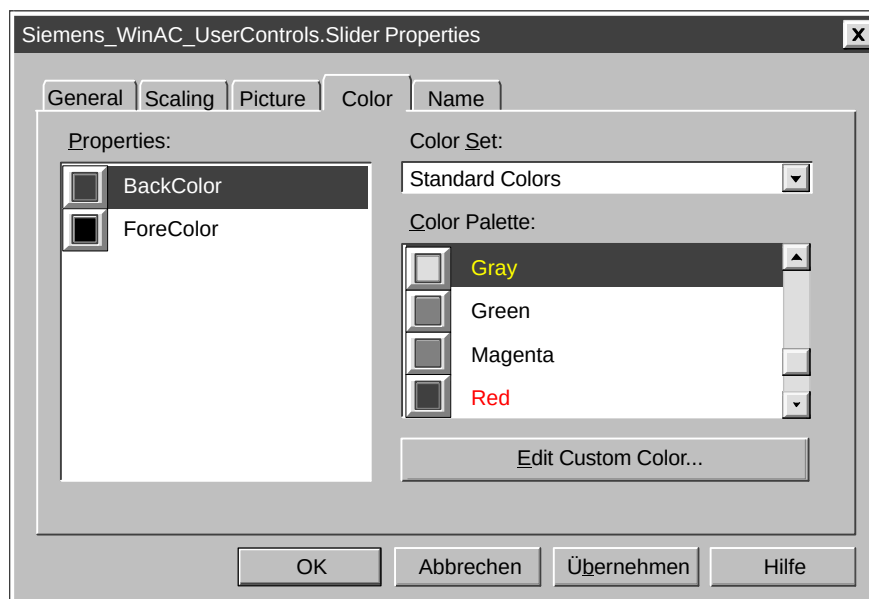


Bild 6-23 Eigenschaften des Slider Control (Register "Color")

Hinweis

Die Eigenschaft BackColor kann nur geändert werden, wenn Sie im Register "General" Style: Standard ausgewählt haben. Die Eigenschaft ForeColor können Sie sowohl für Style: Standard und für Style: Graphical ändern.

Register "Name"

Im Register "Name" des Dialogfelds "Eigenschaften" (siehe Bild 6-24) können Sie den Controls, die Sie in den Container eingefügt haben, beschreibende Namen zuordnen. Dieses Register ist nur verfügbar, wenn Sie mit Controls im SIMATIC Computing Container arbeiten.

Sie geben den neuen Namen in das Feld "Control Name" ein und bestätigen mit "Übernehmen" oder "OK". Der neue Name wird nun in der Liste "Select Control" in der Funktionsleiste aufgeführt. Durch Doppelklick auf das gewünschte Control rufen Sie die Eigenschaften des Control im Dialogfeld "Eigenschaften" auf.

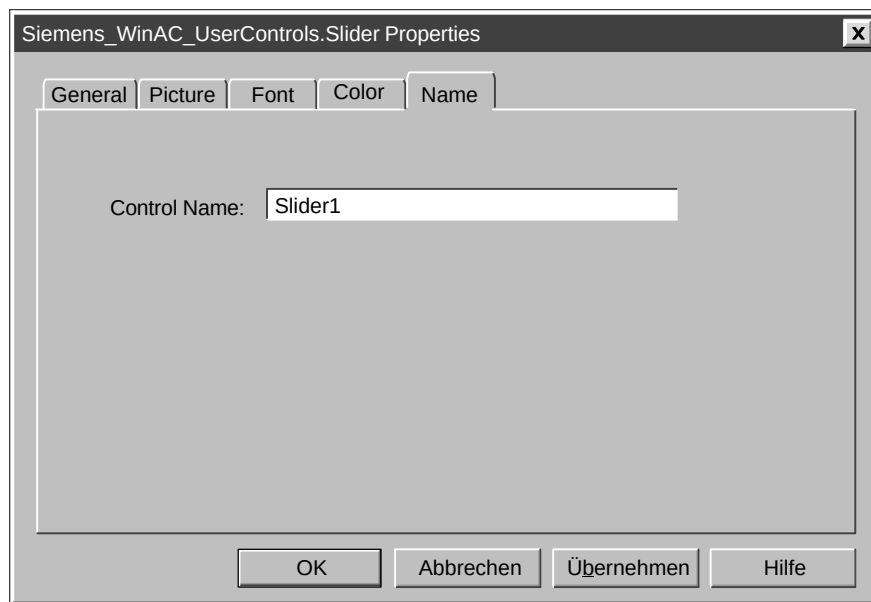


Bild 6-24 Eigenschaften des Slider Control (Register "Name")

6.13 Eigenschaften und Methoden des Slider Control

Mit den in Tabelle 6-9 aufgeführten Eigenschaften und Methoden können Sie das Slider Control bearbeiten.

Tabelle 6-9 Eigenschaften und Methoden des Slider Control

Eigenschaft / Methode	Beschreibung	Seite
Methode AboutBox	Zeigt das Dialogfeld "Info" des Control an.	B-1
Eigenschaft BackColor	Legt die Hintergrundfarbe des Control fest.	B-5
Eigenschaft Direction	Stellt die Ausrichtung ein (waagrecht oder senkrecht).	B-14
Eigenschaft DisplayValue	Gibt den skalierten Wert für das Control aus.	B-18
Eigenschaft Enabled	Legt fest, ob das Control auf vom Anwender erzeugte Ereignisse reagiert.	B-19
Eigenschaft Factor	Gibt den Skalierfaktor an, der benötigt wird, wenn Sie mit Hilfe einer Formel skalieren möchten (in Zusammenhang mit der Eigenschaft ScaleMode).	B-20
Eigenschaft ForeColor	Legt die Farbe für den Vordergrund des Control fest.	B-23
Eigenschaft KnobHeight	Legt die Höhe der vom Control angezeigten Anzeige fest.	B-24
Eigenschaft KnobPicture	Legt das graphische Element (Bild) für die Anzeige auf dem Control fest.	B-24
Eigenschaft KnobWidth	Legt die Breite der vom Control angezeigten Anzeige fest.	B-25
Eigenschaft LargeChange	Legt fest, wie weit sich die Bildlaufanzeige im Bildlauf-feld bewegt, wenn das Control aktiviert ist und Sie eine der Tasten Seite-nach-oben/Seite-nach-unten drücken.	B-25
Eigenschaft Locked	Versetzt das Control in einen schreibgeschützten Zustand. Standardmäßig ist das Control nicht verriegelt, so daß Sie Zahlen eingeben können.	B-26
Eigenschaft Max	Gibt den höchsten skalierten Wert des Control an, wenn die Eigenschaft ScaleMode auf wByRange oder wScaleNone gesetzt ist.	B-26
Eigenschaft Min	Gibt den niedrigsten skalierten Wert des Control an, wenn die Eigenschaft ScaleMode auf wByRange oder wScaleNone gesetzt ist.	B-26
Eigenschaft Offset	Gibt den Offset an, der benötigt wird, wenn Sie mit Hilfe einer Formel skalieren möchten (in Zusammenhang mit der Eigenschaft ScaleMode).	B-27
Eigenschaft Picture	Legt das graphische Element (Bild) für das Control fest.	B-28
Eigenschaft RawMax	Gibt den höchsten skalierten Rohwert des Control an (wenn die Eigenschaft ScaleMode auf wByRange gesetzt ist).	B-32
Eigenschaft RawMin	Definiert den unteren Wert des Quellbereichs zum Skalieren eines Werts. Die Eigenschaft ScaleMode muß auf "wByRange" gesetzt sein.	B-32

Tabelle 6-9 Eigenschaften und Methoden des Slider Control, Fortsetzung

Eigenschaft / Methode	Beschreibung	Seite
Eigenschaft ScaleMode	Gibt an, auf welche Art Werte skaliert werden sollen.	B-34
Eigenschaft ShowMinMax	Gibt an, ob das Control den Wertebereich (Minimum und Maximum) anzeigt.	B-36
Eigenschaft Style	Stellt die Darstellungsart (standard oder graphisch) des Control ein.	B-39
Eigenschaft SmallChange	Legt fest, wie weit sich die Bildlaufanzeige im Bildlauf-feld bewegt, wenn das Control aktiviert ist und Sie eine der Richtungstasten nach oben/unten oder nach rechts/links drücken.	B-37
Eigenschaft StretchMode	Gibt den Stretch-Modus des graphischen Elements des Control an.	B-38
Eigenschaft Ticks	Stellt die Anzahl der Teilstriche ein.	B-40
Eigenschaft Value	Enthält den Wert, der mit der Control Engine verknüpft ist.	B-43

6.14 Ereignisse des Slider Control

Das Control reagiert auf die in Tabelle 6-10 aufgeführten Ereignisse.

Tabelle 6-10 Ereignisse des Slider Control

Ereignis	Beschreibung	Seite
Ereignis Change	Tritt auf, wenn sich der Wert der Eigenschaft Value ändert.	C-1
Ereignis Click	Tritt auf, wenn sich der Mauszeiger auf dem Control befindet und eine Maustaste gedrückt und wieder losgelassen wird.	C-1
Ereignis DblClick	Tritt auf, wenn sich der Mauszeiger auf dem Control befindet und mit einer Maustaste doppelt geklickt wird.	C-2
Ereignis Error	Tritt auf, wenn eine Eigenschaft auf einen unzulässigen Wert gesetzt ist.	C-2
Ereignis KeyDown	Tritt auf, wenn Sie eine Taste drücken, während ein Control aktiviert bzw. im Fokus ist.	C-3
Ereignis KeyPress	Tritt auf, wenn eine ANSI-Taste gedrückt und wieder losgelassen wird, während das Control aktiviert bzw. im Fokus ist.	C-5
Ereignis KeyUp	Tritt auf, wenn eine Taste gedrückt und wieder losgelassen wird, während das Control aktiviert bzw. im Fokus ist.	C-5
Ereignis MouseDown	Tritt auf, wenn sich der Mauszeiger auf dem Control befindet und die Maustaste gedrückt wird.	C-6
Ereignis MouseMove	Tritt auf, wenn Sie den Mauszeiger über ein Control bewegen.	C-7
Ereignis MouseUp	Tritt auf, wenn sich der Mauszeiger auf dem Control befindet und die Maustaste losgelassen wird.	C-8

S7 Diagnostic Buffer Control (DBuffer)

7

Der Diagnosepuffer der S7-Automatisierungssysteme ist ein Ringpuffer, der Einträge enthält, die vom Betriebssystem der S7-Automatisierungssysteme geschrieben werden. Jeder Eintrag enthält Informationen zu einem bestimmten Diagnoseereignis. Diese Ereignisse werden in der Reihenfolge angezeigt, in der sie erzeugt wurden, wobei das jüngste Ereignis an erster Stelle in der Liste steht. Das DBuffer Control gewährt Ihrem Programm Zugriff auf den Diagnosepuffer und es zeigt die Ereignisse an.

Ausführliche Informationen zum Diagnosepuffer entnehmen Sie der Online-Hilfe in der Programmiersoftware STEP 7 und der Dokumentation der S7-Automatisierungssysteme.

Kapitel	Beschreibung	Seite
7.1	Arbeiten mit dem S7-Diagnosepuffer	7-2
7.2	Konfigurieren des DBuffer Control	7-4
7.3	Eigenschaften, Methoden und Ereignisse des DBuffer Control	7-7

7.1 Zugreifen auf den S7-Diagnosepuffer

Wie Sie in Bild 7-1 sehen, können Sie mit dem DBuffer Control auf den Diagnosepuffer eines S7-Automatisierungssystems zugreifen, ohne hierzu STEP 7 oder das Data Control einsetzen zu müssen. Das DBuffer Control zeigt folgende Informationen aus dem Diagnosepuffer an:

- Das Control führt die Diagnoseereignisse, die vom Controller generiert wurden, mit dem Datum und der Zeit, zu denen das Ereignis aufgetreten ist, an. Sie können die Ereignisse in der Liste anwählen, um sich weitere Informationen anzeigen zu lassen.
- Das Control verfügt über ausführliche Angaben zu jedem Ereignis. Sie können sich anzeigen lassen:
 - Eine textuelle Beschreibung der Informationen, z.B. Adresse der Operation, die das Ereignis ausgelöst hat sowie Betriebszustandswechsel, der vom Ereignis verursacht wurde.
 - Hexadezimalwerte für die 20 Bytes des Diagnoseereignisses.
- Das Control zeigt die hexadezimale Ereignis-ID an.

Wenn Sie das Control aktivieren, um die Ereignisse sortieren zu können, haben Sie die Möglichkeit, die Ereignisse mit den Schaltflächen im oberen Teilfenster nach der Art des Ereignisses (entweder nach der Beschreibung oder nach der hexadezimalen Ereignis-ID) oder nach der Reihenfolge ihres Auftretens (dies wird durch die Zeit und das Datum, zu denen das Ereignis generiert wurde, festgelegt) zu sortieren.

Das DBuffer Control benötigt das Data Control nicht, um auf die Control Engine zuzugreifen. Sie richten die Eigenschaften des DBuffer Control für die Control Engine und für andere Eigenschaften ein.

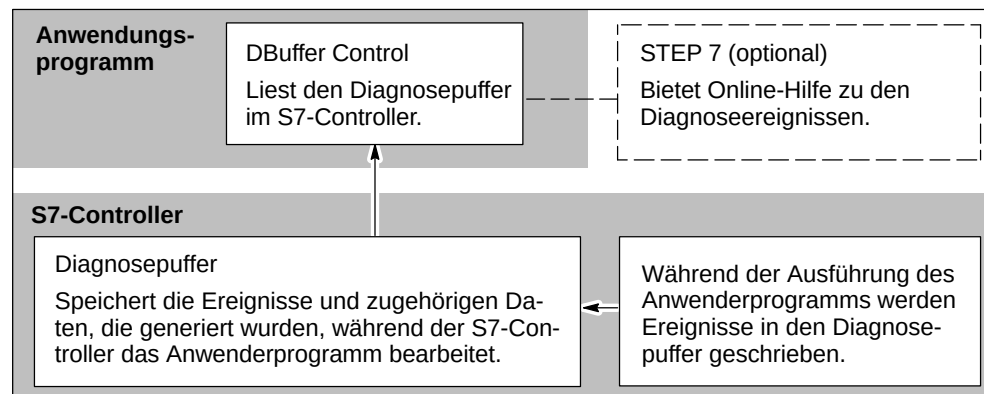


Bild 7-1 Zugreifen auf den Diagnosepuffer eines S7-Controllers

Hinweis

Das DBuffer Control liest und aktualisiert den Diagnosepuffer im Controller nicht automatisch: Sie müssen dies manuell anfordern, indem Sie auf die Schaltfläche "Aktualisieren" klicken.

Wie Sie in Bild 7-2 sehen, bietet das DBuffer Control folgende Elemente:

- Oberes Teilfenster: Führt die Ereignisse aus dem Diagnosepuffer des Controllers auf, einschließlich Nummer (beginnend bei dem jüngsten Ereignis), Datum und Zeit des Ereignisses sowie einer kurzen Beschreibung.
- Unteres Teilfenster: Führt eine ausführliche Beschreibung eines im oberen Teilfenster markierten Ereignisses auf. Diese Angaben umfassen den Namen und die Nummer des Ereignisses, weitere Angaben (z.B. die Adresse der Operation, die das Ereignis ausgelöst hat) und den aktuellen Zustand des Ereignisses.
- Schaltfläche zum Aktualisieren: Liest den Diagnosepuffer und aktualisiert die vom Control angezeigten Angaben. Das DBuffer Control liest den S7-Diagnosepuffer und aktualisiert die Ereignisse nicht automatisch.
- Sprache: Stellt die Sprache (Deutsch, Englisch, Französisch, Italienisch oder Spanisch) für die Beschreibung der Ereignisse ein. Die Sprache für die Spaltenköpfe und die Schaltflächen (Deutsch, Englisch oder Französisch) wird über das Panel Control oder den Container eingestellt.
- Schaltfläche für Hilfe zu Ereignissen: Zeigt die Online-Hilfe (aus STEP 7) für das markierte Diagnoseereignis an.

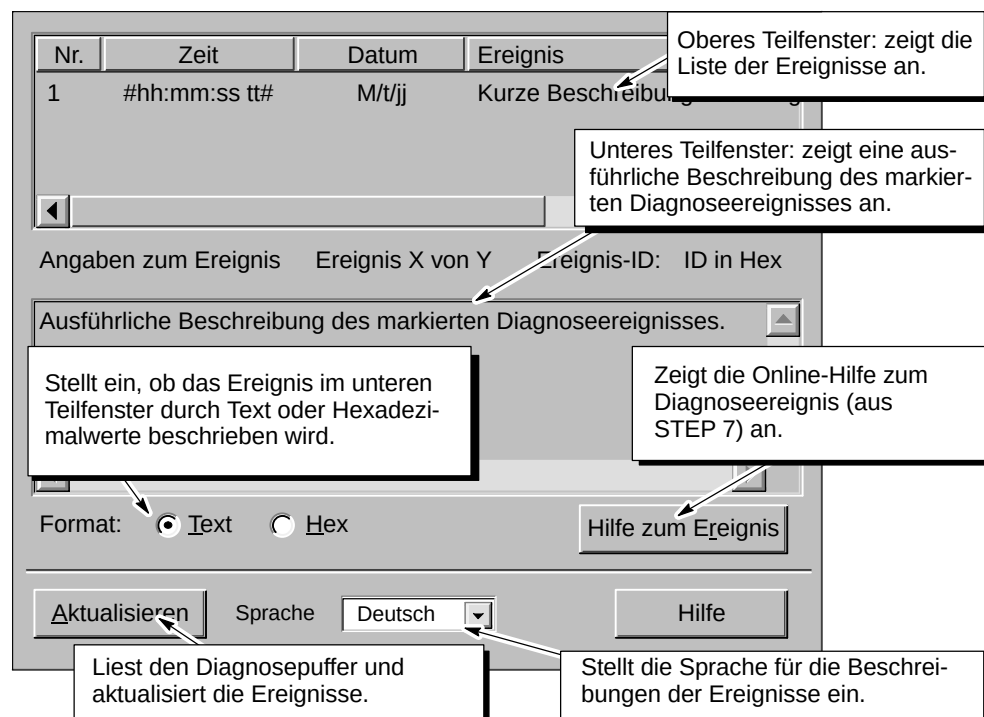


Bild 7-2 Elemente des DBuffer Control

7.2 Konfigurieren des DBuffer Control

Bild 7-3 zeigt das Dialogfeld zum Konfigurieren des Dialogfelds "Eigenschaften" des DBuffer Control. Zum Aufrufen des Dialogfelds "Eigenschaften" klicken Sie mit der rechten Maustaste und wählen den Befehl **Eigenschaften DB1**. Sie können auf die Eigenschaften des Control auch mit Ihrem Anwendungsprogramm zugreifen. Im Register "Diagnostic Buffer" können Sie folgende Eigenschaften einstellen:

- **Enable Sort:** Ist dieses Kontrollkästchen aktiviert, kann der Anwender die Diagnoseereignisse sortieren, indem er auf die Kopfzeilen der Spalten des Diagnosepuffers (Nr., Zeit, Datum oder Ereignis) klickt (siehe Bild 7-2).
- **Text oder Hex(adezimal):** Diese Eigenschaft (FormatDisplay) zeigt die Informationen zum Diagnoseereignis entweder als Hexadezimalzahlen oder als Text an.
- **Control Engine:** Diese Eigenschaft (ControlEngine) wählt die Control Engine. Für WinLC geben Sie **WinLC** ein und Sie geben **wsS7=3** für eine Slot-PLC wie die CPU 416-2 DP ISA ein.

Sie können die Verbindung zur Control Engine auch testen, indem Sie auf die Schaltfläche "Connection Test" klicken.

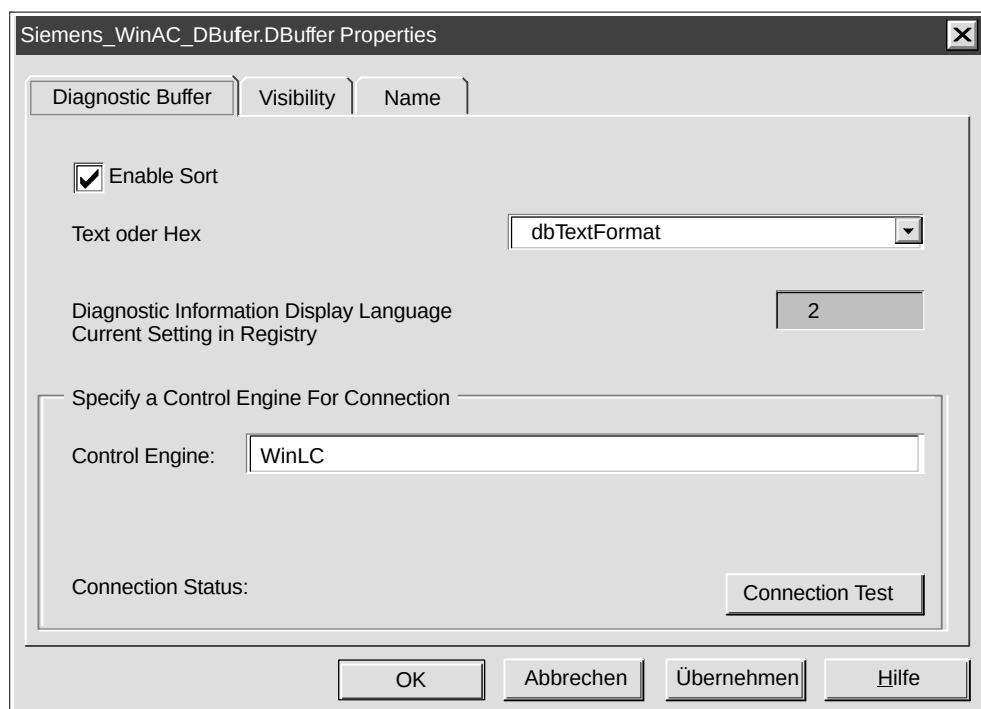


Bild 7-3 Register "Diagnostics Buffer" für das DBuffer Control

Bild 7-4 zeigt das Register "Visibility" im Dialogfeld "Eigenschaften" des DBuffer Control. In diesem Register können Sie die verschiedenen Elemente des DBuffer Control auswählen, die angezeigt werden sollen:

- Display Upper Panel: Ist die Eigenschaft DisplayUpperPanel aktiviert, wird das obere Teilfenster des Diagnosepuffers angezeigt. Dieses Teilfenster zeigt die Liste des Diagnosepuffers an.
- Display Lower Panel: Ist die Eigenschaft DisplayLowerPanel aktiviert, wird das untere Teilfenster des Diagnosepuffers angezeigt. Dieses Teilfenster führt eine ausführliche Beschreibung eines im oberen Teilfenster markierten Ereignisses auf.
- Display Format Buttons: Ist die Eigenschaft DisplayFormatButtons aktiviert, werden Schaltflächen angezeigt, mit denen eingestellt werden kann, ob die Angaben zum Ereignis als Text oder Hexadezimalwert angezeigt werden.
- Display Help-On-Event Button: Ist die Eigenschaft DisplayHelpOnEvent aktiviert, wird die Schaltfläche "Hilfe zum Ereignis" angezeigt, mit der die Online-Hilfe zum markierten Diagnoseereignis aufgerufen werden kann. (Hierzu muß STEP 7 auf dem gleichen PC wie das DBuffer Control installiert sein.)
- Display Help Button: Ist die Eigenschaft DisplayHelpButton aktiviert, wird die Schaltfläche "Hilfe" angezeigt, mit der die Online-Hilfe zum DBuffer Control aufgerufen werden kann.
- Display Update Button: Ist die Eigenschaft DisplayUpdateButton aktiviert, wird die Schaltfläche "Aktualisieren" angezeigt, die den Diagnosepuffer im S7-Controller liest.

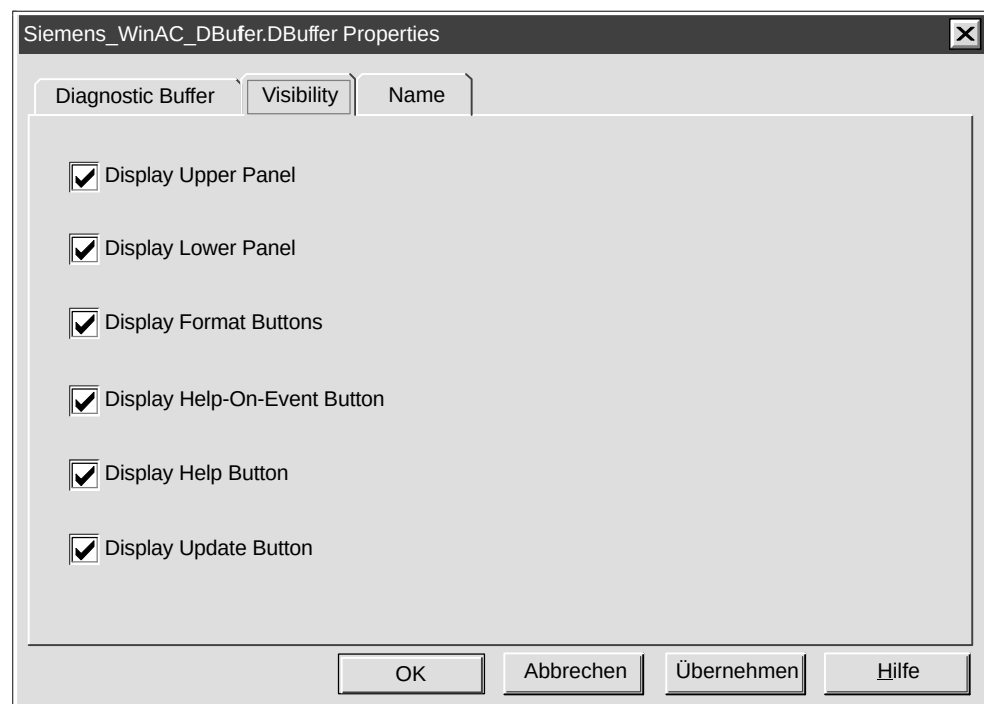


Bild 7-4 Register "Visibility" des DBuffer Control

Register "Name"

Im Register "Name" des Dialogfelds "Eigenschaften" (siehe Bild 7-5) können Sie den Controls, die Sie in den Container eingefügt haben, beschreibende Namen zuordnen. Dieses Register ist nur verfügbar, wenn Sie mit Controls im SIMATIC Computing Container arbeiten.

Sie geben den neuen Namen in das Feld "Control Name" ein und bestätigen mit "Übernehmen" oder "OK". Der neue Name wird nun in der Liste "Select Control" in der Funktionsleiste aufgeführt. Durch Doppelklick auf das gewünschte Control rufen Sie die Eigenschaften des Controls im Dialogfeld "Eigenschaften" auf.

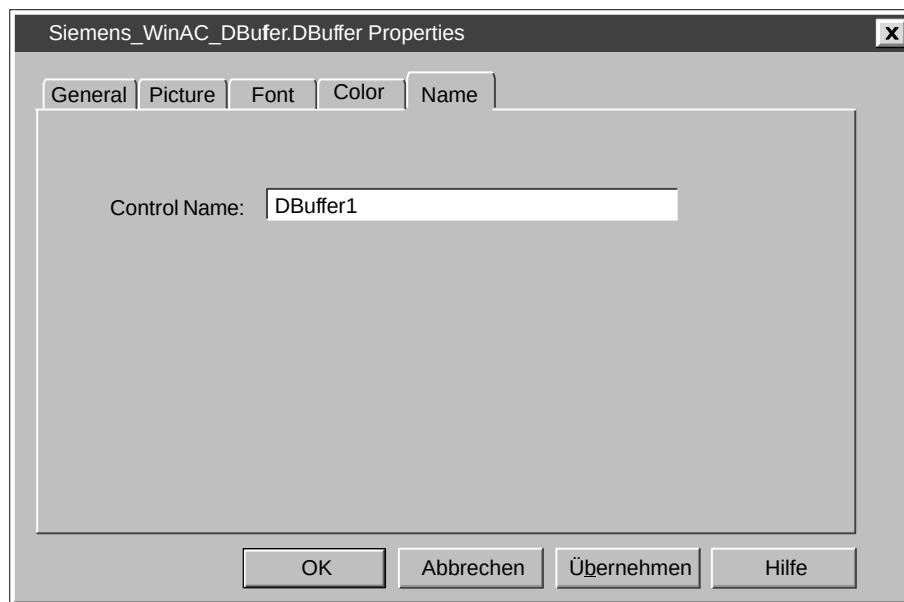


Bild 7-5 Dialogfeld "Eigenschaften" des DBuffer Control (Register "Name")

7.3 Eigenschaften und Methoden des DBuffer Control

Mit den in Tabelle 7-1 aufgeführten Eigenschaften und Methoden können Sie das DBuffer Control bearbeiten.

Tabelle 7-1 Eigenschaften und Methoden des DBuffer Control

Eigenschaft oder Methode	Beschreibung	Seite
Eigenschaft bDiagBuffOK	Prüft die Verbindung zum Diagnosepuffer.	B-5
Eigenschaft bEngineConnected	Prüft die Verbindung zur Control Engine.	B-6
Eigenschaft ControlEngine	Speichert den Dateinamen der Control Engine.	B-10
Eigenschaft DisplayFormatButtons	Zeigt die Schaltflächen "Text" und "Hexadezimal" an bzw. blendet sie aus.	B-15
Eigenschaft DisplayHelpButton	Zeigt die Schaltfläche "Hilfe" an bzw. blendet sie aus.	B-15
Eigenschaft DisplayHelpOnEventButton	Zeigt die Schaltfläche "Hilfe zum Ereignis" an bzw. blendet sie aus.	B-16
Eigenschaft DisplayLowerPanel	Zeigt das untere Teilfenster des DBuffer Control an bzw. blendet es aus.	B-17
Eigenschaft DisplayUpdateButton	Zeigt die Schaltfläche "Aktualisieren" an bzw. blendet sie aus.	B-17
Eigenschaft DisplayUpperPanel	Zeigt das obere Teilfenster des DBuffer Control an bzw. blendet es aus.	B-18
Eigenschaft EnableSort	Aktiviert oder deaktiviert das Sortieren der Spalten im oberen Teilfenster des DBuffer Control.	B-19
Eigenschaft FormatDisplay	Ändert die Formatierung der zusätzlichen Informationen zu einem bestimmten Ereignis.	B-23
Methode PopUpHelp	Zeigt die Online-Hilfe für das DBuffer Control an.	B-29
Methode PopUpHelpOnEvent	Zeigt die Online-Hilfe (aus STEP 7) für das markierte Diagnoseereignis an.	B-29
Methode SelectEvent	Markiert ein bestimmtes Ereignis im oberen Teilfenster des DBuffer Control.	B-35
Methode Update	Liest den Diagnosepuffer in der Control Engine und aktualisiert die Ereignisse, die das Control auflistet.	B-42

Entwerfen von einfachen Prozeß- Formularen im WinAC SoftContainer

8

Kapitelübersicht

SIMATIC Computing verfügt über einen OLE-Container (SoftContainer) zum Empfangen und Anzeigen der Daten aus der Control Engine. In diesen Container können Sie Ihre eigenen Fremd-Controls und die SIMATIC Controls in ein Prozeß-Formular einfügen.

In diesem Kapitel erfahren Sie, wie Sie die Controls in den Container einfügen und anordnen. Ausführliche Informationen zu bestimmten SIMATIC Controls entnehmen Sie den folgenden Kapiteln:

- Informationen zum Data Control entnehmen Sie Kapitel 5.
- Informationen zum Diagnostic Buffer Control (DBuffer) entnehmen Sie Kapitel 7.
- Informationen zu den anderen SIMATIC Controls (Button Control, Edit Control, Label Control und Slider Control) entnehmen Sie den Kapiteln 6, 7, 8 und 9.

Kapitel	Beschreibung	Seite
8.1	Starten des SIMATIC Computing SoftContainer	8-2
8.2	Erstellen eines Prozeß-Formulars	8-4
8.3	Wechseln von der Betriebsart Design in die Betriebsart RUN	8-6
8.4	Speichern Ihres Prozeß-Formulars	8-8

8.1 Starten des SIMATIC Computing SoftContainer

SIMATIC Computing umfaßt einen Container für die verschiedenen SIMATIC Controls. Zum Anlegen eines Containers wählen Sie im Windows Startmenü den Befehl **Simatic > PC Based Control > SIMATIC Computing Softcontainer**. Sie können auch mit der Maus auf das Symbol von SIMATIC Computing doppelklicken. In Bild 8-1 sehen Sie ein Beispiel für einen Container mit folgenden Elementen:

- Die Funktionsleiste enthält Schaltflächen zum schnellen Aufrufen häufig verwendeter Funktionen (z.B. zum Öffnen von Prozeß-Formularen, zum Ausschneiden und Einfügen). Sie enthält außerdem die Symbole für die SIMATIC Controls, die in SIMATIC Computing zur Verfügung gestellt werden.
- Die Funktionsleiste enthält auch ein Feld, das den Namen des ausgewählten Control anzeigt. In dem aufklappbaren Listenfeld können Sie die Controls im Prozeß-Formular auswählen.
- Die Statuszeile zeigt Informationen zur Betriebsart des S7Soft Containers an (Design oder RUN). Außerdem zeigt die Statuszeile an, welches Control ausgewählt ist, einschließlich Größe und Position im Prozeß-Formular.
- Ein leeres Prozeß-Formular (S7Soft1), in das Controls eingefügt werden können.

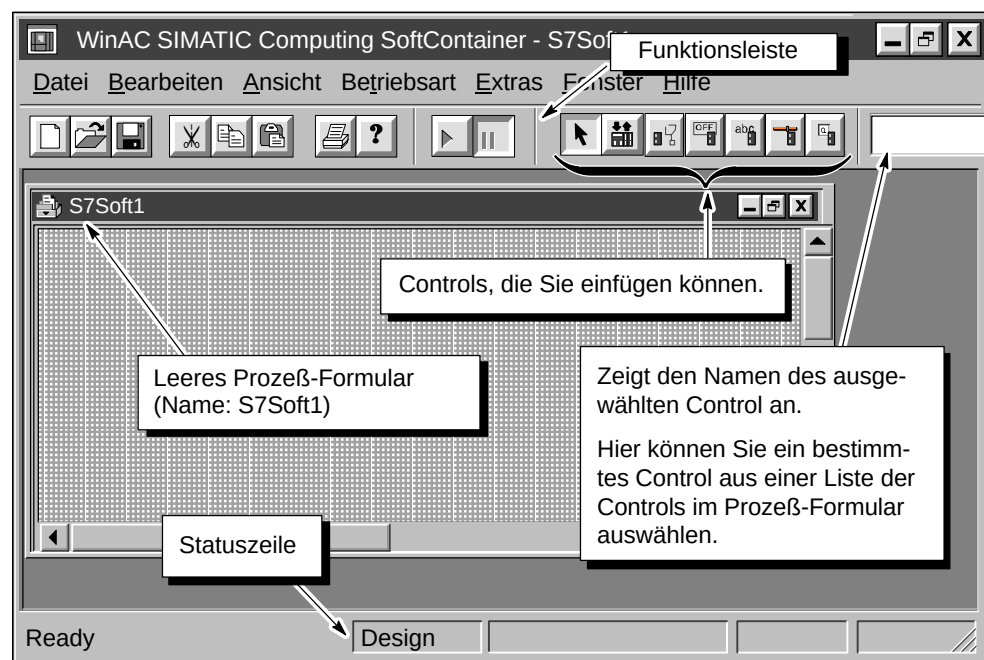


Bild 8-1 Container mit einem leeren Prozeß-Formular

Arbeiten mit dem Gitternetz und der Statuszeile

Der Container verfügt über ein Gitternetz, das das Positionieren und Ändern der Größe der Controls erleichtert. In Bild 8-2 sehen Sie, daß die Statuszeile Informationen zum markierten Control angibt:

- Positionsangaben. Die Statuszeile zeigt die aktuellen Koordinaten des Controls im Gitternetz an. (Diese Angaben werden auch dann angezeigt, wenn das Gitternetz deaktiviert ist.)
- Größenangaben. Die Statuszeile zeigt die Größe des Controls an (Breite x Höhe).

Wenn Sie mehrere Controls auswählen, können Sie mit Hilfe der Angaben in der Statuszeile die Größe bzw. die Position der Controls anpassen.

Die Statuszeile und das Gitternetz können Sie ein- und ausschalten.

- Zum Aktivieren des Gitternetzes wählen Sie den Menübefehl **Ansicht > Gitternetz**. Das Gitternetz ist aktiviert, wenn der Menübefehl mit einem Häkchen markiert ist.
- Zum Anzeigen der Statuszeile wählen Sie den Menübefehl **Ansicht > Statuszeile**. Die Statuszeile wird angezeigt, wenn der Menübefehl mit einem Häkchen markiert ist.

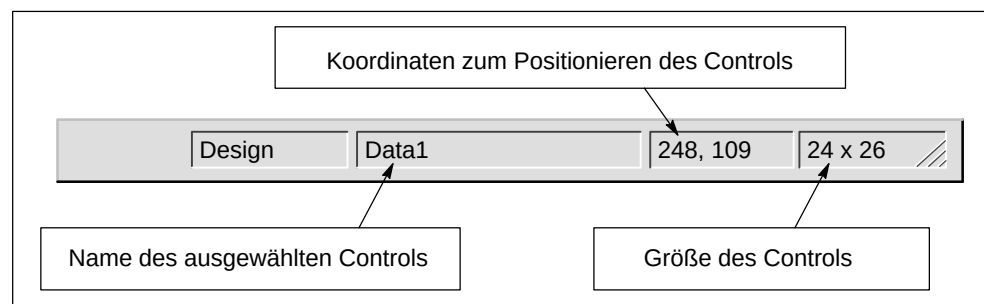


Bild 8-2 Elemente in der Statuszeile

8.2 Erstellen eines Prozeß-Formulars

Ein "Prozeß-Formular" ist ein Dokument bzw. eine Datei, die Sie mit dem WinAC SoftContainer erstellt haben. Es enthält ActiveX Controls, mit denen Sie die Daten der Control Engine überwachen und ändern können.

Einfügen eines SIMATIC Control in ein Prozeß-Formular

Gehen Sie folgendermaßen vor, um ein SIMATIC Control in Ihr Prozeß-Formular einzufügen (siehe Bild 8-3):

1. Klicken Sie in der Funktionsleiste auf das Symbol des Controls, das Sie einfügen möchten. (Bild 8-3 zeigt, wie ein Data Control in ein Prozeß-Formular eingefügt wird.)
2. Bewegen Sie den Zeiger in das geöffnete Prozeß-Formular. Innerhalb des Prozeß-Formulars verwandelt sich der Mauszeiger in ein Fadenkreuz.
3. Klicken Sie mit der linken Maustaste, um das ausgewählte Control einzufügen.

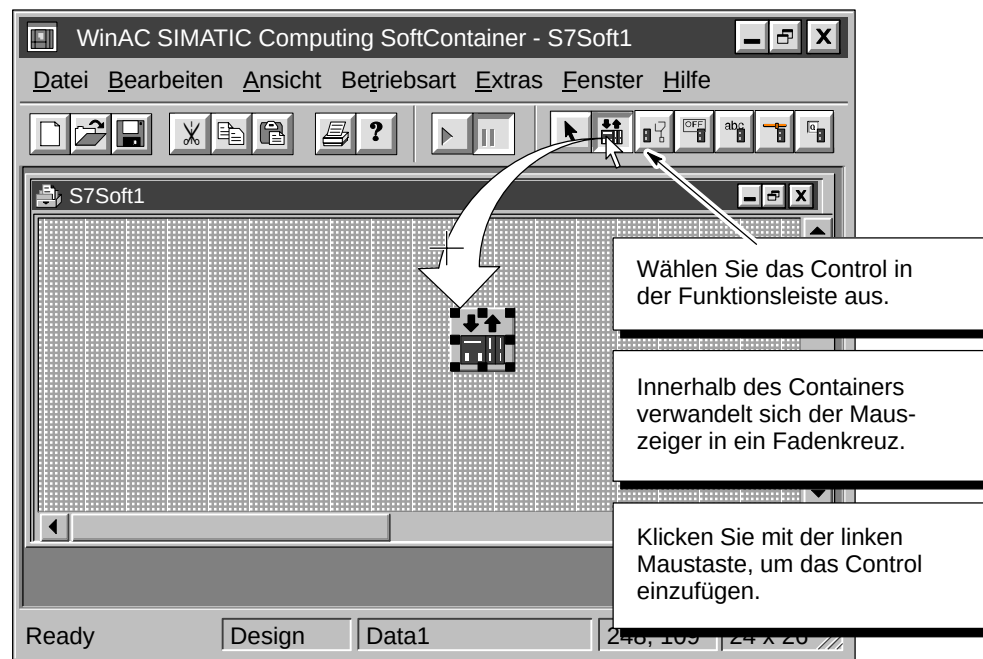


Bild 8-3 Einfügen eines Controls über die Funktionsleiste

Einfügen von Fremd-Controls in ein Prozeß-Formular

Sie können in Ihr Prozeß-Formular neben den SIMATIC Controls auch andere ActiveX Controls einfügen. Gehen Sie folgendermaßen vor, wenn Sie ein anwenderspezifisches Control oder ein Fremd-Control in Ihr Prozeß-Formular einfügen möchten:

1. Wählen Sie den Menübefehl **Bearbeiten > Control einfügen**. Daraufhin wird das Dialogfeld "Control einfügen" angezeigt.
2. Wählen Sie ein anwenderspezifisches Control oder ein Fremd-Control aus, das in das Prozeß-Formular aufgenommen werden soll (siehe Bild 8-4). (Sie können für das Control ein Symbol in die Funktionsleiste des SoftContainer aufnehmen, wenn Sie das Kontrollkästchen "Control in Funktionsleiste aufnehmen" aktivieren. Dann können Sie weitere Controls über das Symbol in der Funktionsleiste in Ihr Prozeß-Formular aufnehmen.)
3. Bestätigen Sie mit "OK" und fügen Sie das Control in das Prozeß-Formular ein.

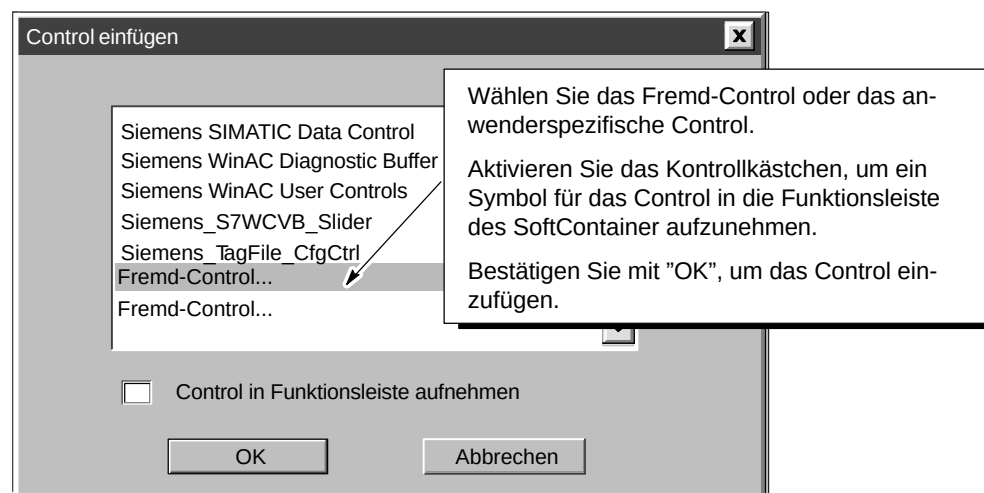


Bild 8-4 Einfügen eines Fremd-Control in ein Prozeß-Formular

Konfigurieren des Data Control

Bevor Sie eine Verbindung zur Control Engine herstellen können, müssen Sie das Data Control für die Kommunikation mit der Control Engine einrichten. Wie Sie das Data Control einrichten erfahren Sie in Kapitel 5.

Hinweis

Das Diagnostic Buffer Control stellt die Verbindung zum Automatisierungssystem nicht über das Data Control her.

8.3 Wechseln von der Betriebsart Design in die Betriebsart RUN

Wenn Sie den SoftContainer von der Betriebsart Design in die Betriebsart RUN versetzen, verbinden Sie die Controls mit der Control Engine. Diese Betriebsarten definieren nur die Funktionsweise des SoftContainers. Sie wirken sich nicht auf die Betriebszustände der Control Engine aus.



Warnung

Nachdem Sie ein SIMATIC oder Fremd-Control mit den Prozeßdaten verbunden haben, indem Sie der Eigenschaft Value des Controls eine Variable zugeordnet haben, wirken sich alle Änderungen, die Sie für den vom Control angezeigten Wert eingeben, sofort auf die Prozeßdaten aus.

Durch das Ändern von Prozeßdaten können Sie unvorhersehbare Reaktionen der Geräte im Prozeß auslösen, die zu Tod, schwerer Körperverletzung und/oder Sachschaden führen können.

Führen Sie die in diesem Kapitel beschriebenen Vorgehensweisen nicht aus, wenn Ihre Control Engine mit einem tatsächlichen Prozeß verbunden ist. Die hier beschriebenen Vorgehensweisen dienen nur zu Übungszwecken. Ändern Sie keine Daten, die unvorhersehbaren Betrieb der Geräte im Prozeß auslösen könnten. Installieren Sie einen physikalischen NOT-AUS-Schaltkreis für die Maschine bzw. den Prozeß.

Wechseln des Betriebszustands des SoftContainers

Der SoftContainer besitzt zwei Betriebszustände, die nicht mit den Betriebszuständen der Control Engine zu verwechseln sind:

- In der Betriebsart Design können Sie die Controls in Ihr Prozeß-Formular einfügen und ändern. Außerdem können Sie die Eigenschaften einstellen.
- In der Betriebsart RUN wird eine Verbindung zwischen den Controls und der Control Engine hergestellt. Sie können die Werte in der Control Engine ändern, Sie können jedoch nicht die Eigenschaften der Controls ändern.

Hinweis

Die Controls benötigen eine aktive Control Engine (z.B. WinLC), um auf Prozeßdaten zugreifen zu können. Stellen Sie sicher, daß die Control Engine aktiviert ist, bevor Sie den SoftContainer von der Betriebsart Design in die Betriebsart RUN versetzen.

Zum Einstellen der Betriebsart des Containers gehen Sie folgendermaßen vor:

1. Stellen Sie sicher, daß die Control Engine (z.B. WinLC) aktiviert ist. Informationen zum Starten der Control Engine entnehmen Sie der Dokumentation der Control Engine.
2. Wählen Sie die Schaltfläche für die Betriebsart RUN, um von der Betriebsart Design in die Betriebsart RUN zu wechseln (siehe Bild 8-5). Die Statuszeile zeigt an, daß sich der Container in der Betriebsart RUN befindet.

Beachten Sie, daß das Data Control in der Betriebsart RUN nicht sichtbar ist.

In der Betriebsart RUN können Sie mit Hilfe der Controls die Werte in der Control Engine überwachen und ändern. Wenn Sie wieder in die Betriebsart Design zurückkehren möchten, wählen Sie die Schaltfläche für die Betriebsart Design.



Bild 8-5 Versetzen des Containers in die Betriebsart RUN

8.4 Speichern Ihres Prozeß-Formulars

Sie können die von Ihnen erstellten Prozeß-Formulare unter einem beliebigen Namen und in einem beliebigen Verzeichnis speichern. Zum Speichern Ihrer Prozeß-Formulare gehen Sie folgendermaßen vor:

1. Wählen Sie den Menübefehl **Datei > Speichern unter**. Daraufhin wird das Dialogfeld "Speichern unter" angezeigt (siehe Bild 8-6).
2. Geben Sie einen Namen für das Prozeß-Formular ein.
3. Wählen Sie ein Verzeichnis, in dem das Prozeß-Formular abgelegt werden soll. Das voreingestellte Verzeichnis zum Speichern von Prozeß-Formularen ist das Verzeichnis von SIMATIC Computing (WinCP).
4. Wählen Sie die Schaltfläche "Speichern".

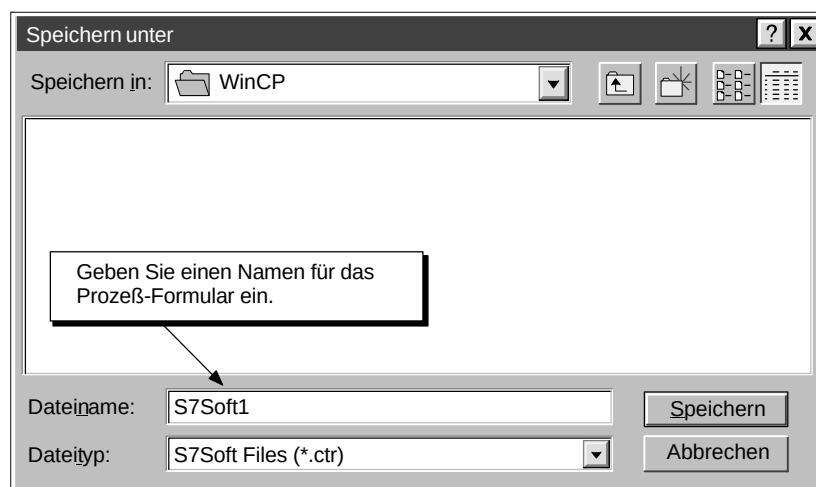


Bild 8-6 Speichern eines Prozeß-Formulars in SIMATIC Computing

Erstellen von Symboldateien mit dem Symboldatei-Konfigurator

9

Der Symboldatei-Konfigurator erstellt Symboldateien, mit deren Hilfe Sie die Adressen, auf die in der Control Engine zugegriffen werden soll, als symbolische Namen angeben können. Eine Symboldatei (*.tsd) ist eine Quelldatei mit Symbolinformationen zu Adressen im Speicher sowie zu Control Engines. Durch die Zuordnung einer Symboldatei können Sie mit symbolischen Namen anstelle von absoluten Adressen arbeiten, wenn Sie im Data Control Variablen zuweisen.

Mehrere STEP 7 Programme können in einer einzigen Symboldatei abgebildet werden, wobei jedes Programm Zugriff auf einen anderen Computer und eine andere Control Engine bietet. So kann SIMATIC Computing auf Daten aus verschiedenen Computern und Control Engines gleichzeitig zugreifen.

Die Symboldatei dient auch als Hilfestellung beim Eingeben von Namen in das Data Control: Wenn Sie "." (Punkt) eingeben, wird eine Liste mit den gültigen Symboldateien angezeigt. Sie können auch über die Schaltfläche "Browse" das Symbol suchen.

Kapitel	Beschreibung	Seite
9.1	Verbinden mehrerer Control Engines über DCOM	9-2
9.2	Zugreifen auf Daten in mehreren Control Engines über Symbole	9-5
9.3	Erstellen einer Symboldatei	9-6
9.4	Konfigurieren einer Symboldatei für lokalen und entfernten Zugriff auf eine Control Engine	9-10
9.5	Ändern des symbolischen Namens der Control Engine im Symboldatei-Editor	9-13

9.1 Verbinden mehrerer Control Engines über DCOM

Wie Sie in Bild 9-1 sehen, können Sie Ihr Programm über DCOM mit Control Engines verbinden, die sich auf unterschiedlichen Computern befinden. Mit dem Symboldatei-Konfigurator erstellen Sie eine Symboldatei, die die symbolischen Namen für die Variablen der verschiedenen Control Engines definiert.

Mehrere STEP 7 Programme können in einer einzigen Symboldatei abgebildet werden, wobei jedes Programm Zugriff auf einen anderen Computer und eine andere Control Engine bietet. So kann das Data Control auf Daten aus verschiedenen Computern und Control Engines gleichzeitig zugreifen.

Dem Abschnitt 9.3 entnehmen Sie Informationen zum Erstellen einer Symboldatei. Dem Abschnitt 9.4 entnehmen Sie Informationen zum Konfigurieren der Control Engine für lokalen und entfernten Zugriff.

Hinweis

Um den Namen eines entfernten Computers in einem STEP 7 Projekt zu erhalten, definieren Sie einen Stationsnamen STEP 7 mit folgenden Komponenten:

- @ (Wenn Sie den Stationsnamen mit “@” beginnen, zeigt dies dem Symboldatei-Konfigurator an, daß sich der Stationsname auf einen entfernten Computer bezieht.)
- Name des entfernten Computers (DCOM-Kennung)

Beispiel (siehe Bild 9-1): **@PC 2**

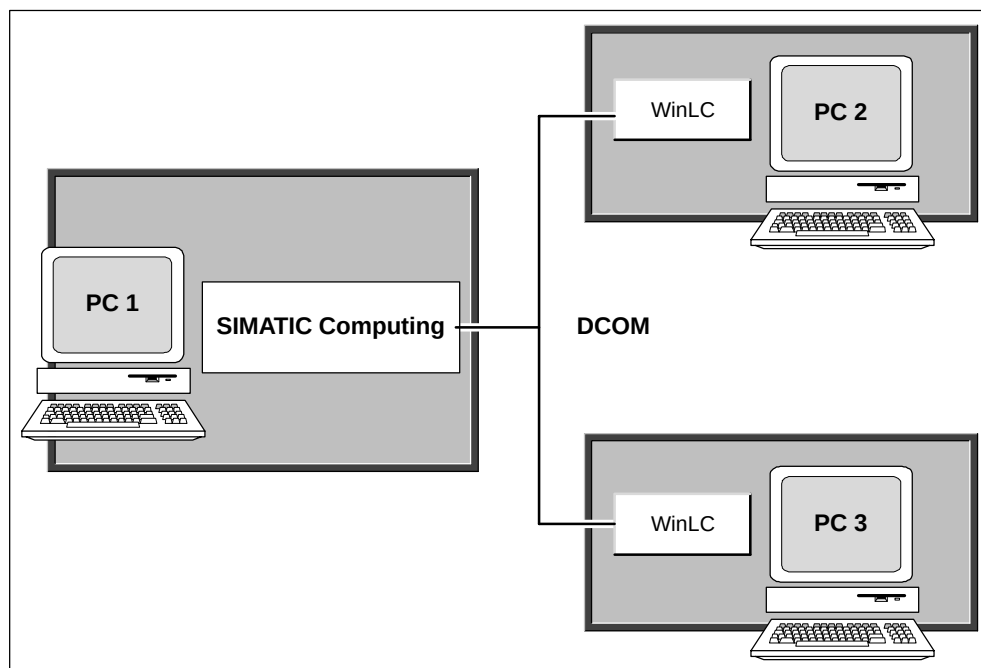


Bild 9-1 Verbinden mehrerer Control Engines über DCOM

Der Symboldatei-Konfigurator erstellt eine Symboldatei, die als Quelle für symbolische Informationen von Adressen im Speicher und von Control Engines dient. Die Symboldatei enthält folgende Informationen:

- **Computername:** Dieser Name kennzeichnet den Computer, auf dem sich die Control Engine befindet, wobei es sich entweder um einen lokalen Computer oder um einen PC im Netz handeln kann. Haben Sie für den Stationsnamen STEP 7 ein Symbol mit dem Zeichen "@" als erstes Zeichen angegeben, erkennt der Symboldatei-Konfigurator den Stationsnamen als DCOM-Adresse für den Computer, auf dem sich die Control Engine befindet.
- **Control Engine:** Hier wird die Control Engine angegeben, auf die das Data Control zugreifen soll. Diese Angabe kommt typischerweise aus der CPU, die in STEP 7 eingerichtet ist. Beispiel: **WinLC** (für WinLC), **wcS7=3** (für eine Slot-PLC wie die CPU 416-2 DP ISA) oder **wcS7=xx, a, b** (für andere Automatisierungssysteme im Netz, wobei xx die Teilnehmeradresse des Automatisierungssystems, a die Nummer des Baugruppenträgers und b die Nummer des Steckplatzes angeben). Ausführliche Informationen zu den Zeichenfolgen der Control Engine entnehmen Sie dem Anhang G.

Erstellen einer Symboldatei mit mehreren Control Engines

Gehen Sie folgendermaßen vor, um eine Symboldatei zu erstellen, die Symbole für verschiedene Control Engines enthält:

1. Rufen Sie über das Windows Startmenü (**Start > Simatic > PC Based Control > WinAC Symboldatei-Konfigurator**) den Symboldatei-Konfigurator auf. Daraufhin wird der Symboldatei-Konfigurator mit einer neuen (leeren) Datei geöffnet.
2. Wählen Sie den Menübefehl **Einfügen > Programm**, um die Programme auszuwählen, die in die Symboldatei aufgenommen werden sollen.
3. Wählen Sie verschiedene Programme aus und nehmen Sie sie in die Symboldatei auf, indem Sie auf die Schaltfläche "->" klicken. Bild 9-2 zeigt zwei Programme ("Master_Mixer" auf PC 2 und "My_Drain" auf PC 3), die eine gemeinsame Symboldatei bilden. Haben Sie die Programme in die Symboldatei aufgenommen, bestätigen Sie mit "OK".
Die Namen der Stationen müssen mit dem Zeichen "@" beginnen (z.B. "@PC 2"), damit der Symboldatei-Konfigurator erkennt, daß die Verbindung zur Control Engine über DCOM verläuft.
4. Mit dem Menübefehl **Datei > Speichern unter...** speichern Sie die Symboldatei (oder benennen sie um).

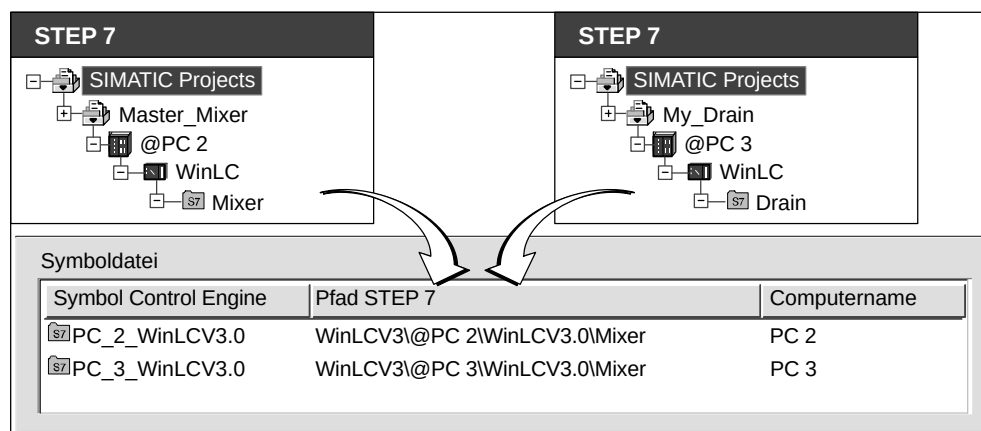


Bild 9-2 Erstellen einer Symboldatei für mehrere Control Engines

Einrichten des Data Control für mehrere Control Engines

Gehen Sie folgendermaßen vor, um das Data Control so einzurichten, daß es eine Symboldatei einsetzt, die Symbole für verschiedene Control Engines enthält:

1. Rufen Sie im Windows Startmenü (**Start > Simatic > PC Based Control > Windows SIMATIC Computing**) den SIMATIC Computing Container auf.
2. Fügen Sie ein Data Control ein.
3. Doppelklicken Sie auf das Data Control (oder wählen Sie den Menübefehl **Bearbeiten > Eigenschaften**), um das Dialogfeld "Eigenschaften" des Data Control aufzurufen.
4. Öffnen Sie das Register "Engine", um die Konfigurationsmöglichkeiten anzuzeigen (siehe Bild 9-3).
5. Wählen Sie die Option "Connection via Tag Source".
6. Klicken Sie auf die Schaltfläche "Blättern" und wählen Sie die Symboldatei, die die Symbole für mehrere Control Engines enthält.
7. Klicken Sie auf die Schaltfläche "OK", um das Data Control für die Kommunikation mit Control Engines, die auf verschiedenen Computern laufen, einzurichten.

Alle Controls, die Sie einfügen und über das Data Control verbinden, greifen auf Variablen in den verschiedenen entfernten Control Engines zu.

Hinweis

Sie müssen die verschiedenen Computer für DCOM eingerichtet haben. In den Abschnitten E.2 und E.3 erfahren Sie, wie Sie die Server- und Client-Computer für DCOM einrichten.

Ausführliche Informationen zu DCOM entnehmen Sie der Online-Hilfe in Windows NT.

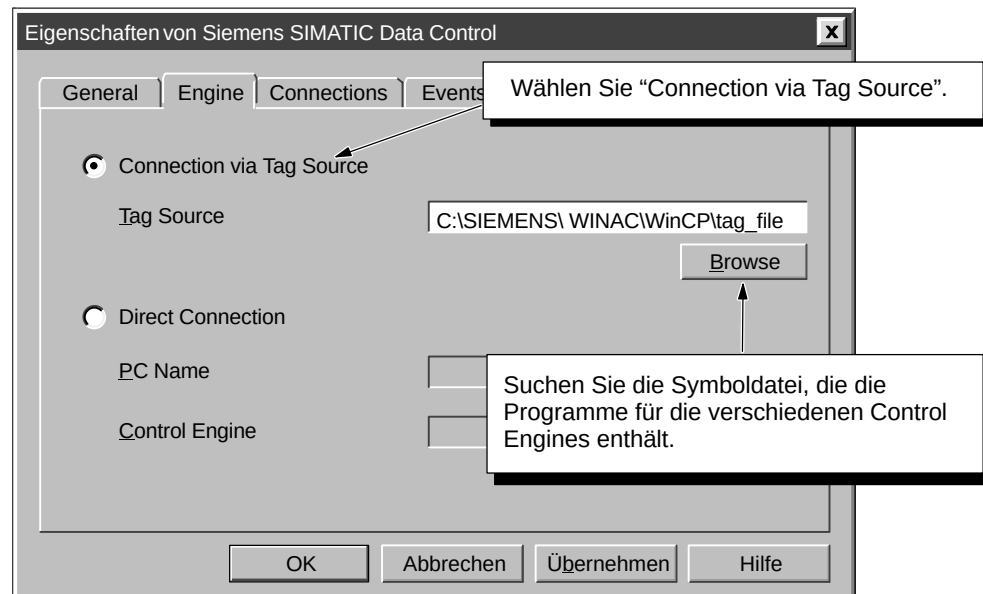


Bild 9-3 Einrichten des Data Control für mehrere Control Engines

9.2 Zugreifen auf Daten in der Control Engine über Symbole

Eine Symboldatei ist eine Quelldatei mit Symbolinformationen zu Adressen im Speicher sowie zu Control Engines. Durch das Verknüpfen einer Symboldatei können Sie symbolische Namen anstelle von absoluten Adressen einsetzen, wenn Sie im Data Control, das mit der Software SIMATIC Computing ausgeliefert wird, Variablen zuweisen. Der Symboldatei-Konfigurator erstellt eine Symboldatei (*.tsd), die als Quelle für symbolische Informationen von Adressen im Speicher und von Control Engines dient (siehe Bild 9-4).

Hinweis

Der Symboldatei-Konfigurator muß auf dem gleichen Computer installiert sein, auf dem sich auch STEP 7 befindet, damit Sie STEP 7 Programme einfügen können, um deren symbolische Adressen zu verwenden. (STEP 7 braucht jedoch nicht installiert zu sein, wenn Sie eine Control Engine ohne symbolische Informationen in die Symboldatei einfügen.)

Damit Sie die Symboldatei mit dem Data Control einsetzen können, muß sich die Symboldatei auf einem Computer befinden, auf den das Data Control Zugriff hat.

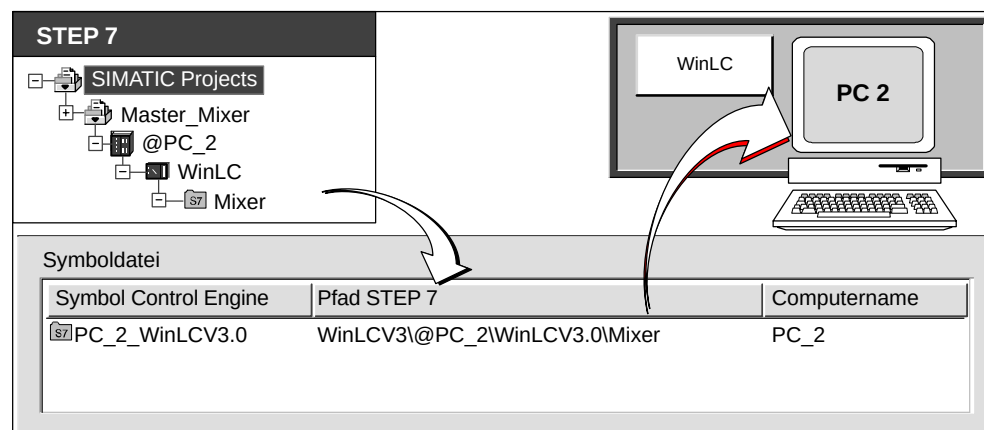


Bild 9-4 Zugreifen auf Daten in der Control Engine über Symbole

Die Symboldatei enthält folgende Elemente:

- **Control Engine:** Hier wird die Control Engine angegeben, auf die das Data Control zugreifen soll. Diese Angabe kommt typischerweise aus der CPU, die in STEP 7 eingerichtet ist. Ausführliche Informationen zu den Zeichenfolgen der Control Engine entnehmen Sie dem Anhang G.
- **Computername:** Dieser Name kennzeichnet den Computer, auf dem sich die Control Engine befindet, wobei es sich entweder um einen lokalen Computer oder um einen PC im Netz handeln kann. Haben Sie für den Stationsnamen STEP 7 ein Symbol mit dem Zeichen "@" als erstes Zeichen angegeben, erkennt der Symboldatei-Konfigurator den Stationsnamen als DCOM-Adresse für den Computer, auf dem sich die Control Engine befindet.

9.3 Erstellen einer Symboldatei

Der Symboldatei-Konfigurator muß auf dem gleichen Computer installiert sein, auf dem sich auch STEP 7 befindet, damit Sie STEP 7 Programme einfügen können, um deren symbolische Adressen zu verwenden. (STEP 7 braucht jedoch nicht installiert zu sein, wenn Sie eine Control Engine ohne symbolische Informationen in die Symboldatei einfügen.) Damit Sie die Symboldatei mit dem Data Control einsetzen können, muß sich die Symboldatei auf einem Computer befinden, auf den das Data Control Zugriff hat.

Dem Abschnitt 9.4 entnehmen Sie Informationen zum Konfigurieren der Control Engine für lokalen und entfernten Zugriff.

Aufrufen des Symboldatei-Konfigurators

Gehen Sie folgendermaßen vor, um den Symboldatei-Konfigurator aufzurufen:

1. Wählen Sie den Menübefehl **Start > Simatic > PC Based Control > WinAC Symboldatei-Konfigurator**.

Daraufhin wird der Symboldatei-Konfigurator mit einer neuen (leeren) Datei geöffnet.

Bild 9-5 zeigt den Symboldatei-Konfigurator mit einer leeren Symboldatei. Sie fügen STEP 7 Programme und Control Engine in die Symboldatei ein.

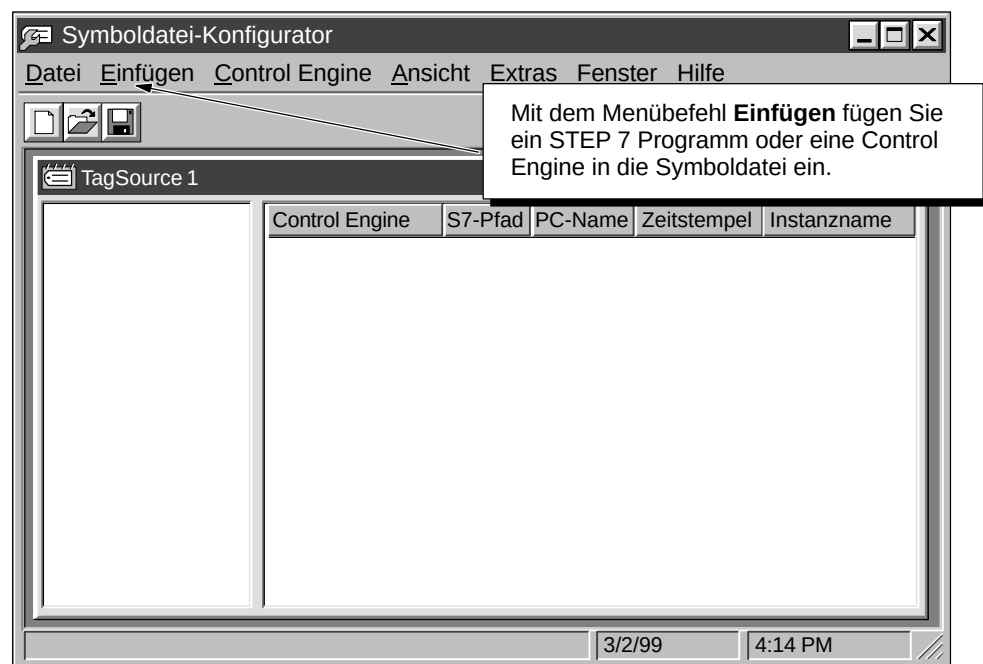


Bild 9-5 Symboldatei-Konfigurator

Einfügen eines Programms oder einer Control Engine in die Symboldatei

Sie können sowohl Control Engines als auch STEP 7 Programme in die Symboldatei einfügen:

- Durch das Einfügen eines Programms mit symbolischer Adressierung können Sie auf die Symbole für die Control Engine zugreifen (STEP 7 "Station"). Es können sich mehrere Stationen (Control Engines) und Programme in der Symboldatei befinden.
- Durch das Einfügen einer Control Engine ohne symbolische Adressierung in die Symboldatei ermöglichen Sie es dem Data Control, auf mehrere Control Engines zuzugreifen. Sie müssen jedoch absolute Adressen verwenden, wenn Sie auf die Daten in diesen Control Engines zugreifen möchten. Durch das Einfügen einer Control Engine werden keine symbolischen Informationen in die Symboldatei geladen.

Gehen Sie folgendermaßen vor, um eine Symboldatei mit symbolischen Daten zu erstellen:

1. Wählen Sie den Menübefehl **Einfügen > Programm**, um das Dialogfeld zum Auswählen der SIMATIC Programme aufzurufen (siehe Bild 9-6).
2. Klicken Sie auf das Symbol "+", um das Projekt, die Station, die CPU und das Programm zu öffnen, das in die Symboldatei eingefügt werden soll (siehe Bild 9-6).
3. Wählen Sie die Schaltfläche "->" (oder doppelklicken Sie auf das Programm), um das Programm in die Symboldatei aufzunehmen.
4. Bestätigen Sie mit "OK".

Nachdem Sie das Programm in die Symboldatei aufgenommen haben, können Sie die Felder "Control Engine" und "Computernamen" bearbeiten. Wenn Sie die Symboltabelle in STEP 7 ändern, können Sie auch die Symboldatei aktualisieren, indem Sie den Menübefehl **Control Engine > Aktualisieren** wählen. Der Konfigurator aktualisiert die Symbole für die Control Engine auf der Basis des Programmpfads.

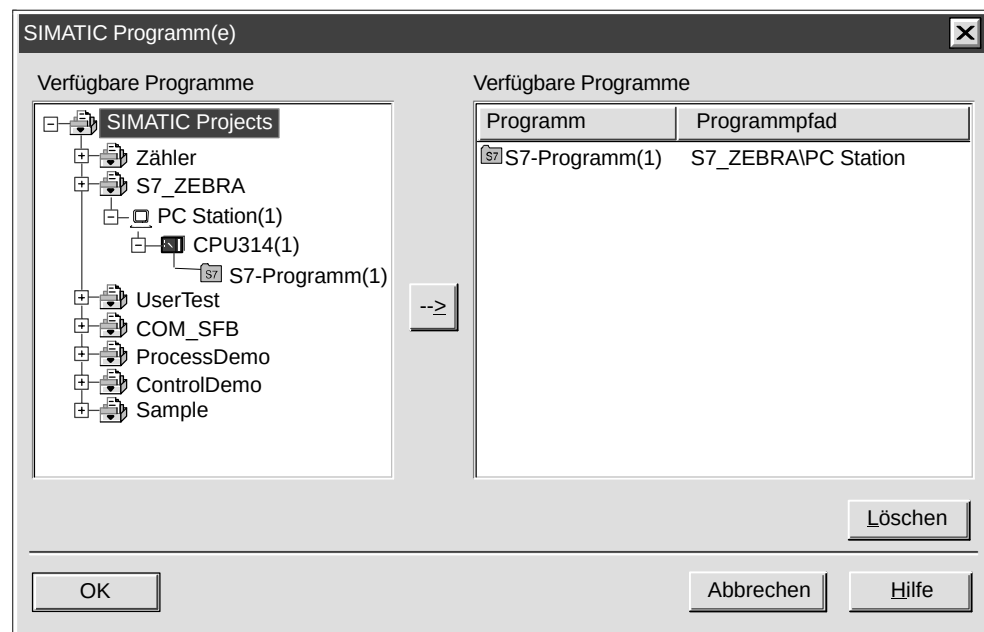


Bild 9-6 Einfügen eines SIMATIC Programms in eine Symboldatei

Verwenden von absoluten Adressen mit einer Symboldatei

Sie können absolute Adressen mit einer Symboldatei verwenden. Absolute Adressen greifen auf die "voreingestellte" Control Engine in der Symboldatei zu. Gehen Sie folgendermaßen vor, um die voreingestellte Control Engine zu kennzeichnen:

1. Wählen Sie in der Dateihierarchie eine Control Engine aus.
2. Kennzeichnen Sie die ausgewählte Control Engine als voreingestellte Control Engine, indem Sie den Menübefehl **Control Engine > Voreinstellen** wählen.
3. Bestätigen Sie dies durch Anklicken der Schaltfläche "Ja".

Ist keine Control Engine in der Symboldatei als voreingestellte Control Engine gekennzeichnet, greifen alle im Data Control eingerichteten absoluten Adressen auf die erste Control Engine in der Datei zu.

9.4 Einrichten einer Symboldatei für lokalen oder entfernten Zugriff auf eine Control Engine

Wenn Sie eine Control Engine in eine Symboldatei einfügen, müssen Sie für die Control Engine angeben, ob sie sich auf einem lokalen Computer (auf dem gleichen Computer wie die Software SIMATIC Computing und die Symboldatei) oder auf einem entfernten Computer (auf den über DCOM zugegriffen wird) befindet.

Wenn Sie die Symbole aus einem STEP 7 Programm für die Control Engine verwenden, kennzeichnet der Symboldatei-Konfigurator die Control Engine mit Hilfe folgender Informationen aus STEP 7:

- **Computername:** Wenn Sie mit dem Symboldatei-Konfigurator eine Control Engine in eine Symboldatei einfügen, ist für die Adresse der Control Engine der "lokale" Computer voreingestellt (wodurch angegeben wird, daß sich die Control Engine auf dem gleichen Computer wie die Software SIMATIC Computing und die Symboldatei befindet).

Wenn Sie Ihr Projekt mit STEP 7 anlegen, können Sie einen Stationsnamen vergeben, der vom Symboldatei-Konfigurator als entfernter Computer ausgewertet wird (wodurch angegeben wird, daß sich die Control Engine auf einem anderen Computer befindet als die Software SIMATIC Computing und die Symboldatei). Um für eine Control Engine anzugeben, daß sie sich auf einem entfernten Computer befindet, vergeben Sie einen Stationsnamen, der mit dem Zeichen "@" beginnt (z.B. @PC 2 oder @PC 3, siehe Beispielkonfiguration in Bild 9-1). Der Symboldatei-Konfigurator erkennt einen Stationsnamen in STEP 7, der mit "@" beginnt, als entfernten Computer.

- **Control Engine:** Der Symboldatei-Konfigurator liest den Typ der Control Engine aus dem STEP 7 Projekt (z.B. WinLC, eine Slot-PLC wie die CPU 416-2 DP ISA oder andere S7-CPU's). Je nach der Art der in STEP 7 eingerichteten CPU erstellt der Symboldatei-Konfigurator eine Kennung für die Control Engine in der Symboldatei. In Anhang G werden die Kennungen der verschiedenen Arten von Control Engines aufgeführt.
- **Symbol Control Engine:** Der Symboldatei-Konfigurator legt ein Symbol für die Control Engine an, indem die folgenden Elemente verbunden und durch einen Unterstrich ("_") voneinander getrennt werden:
 - Computername (der Eintrag im Feld "Computername" im Dialogfeld "Konfiguration Control Engine")
 - STEP 7 Symbol (symbolischer Name für die Control Engine, der in der Symboltabelle angelegt wurde)

Ein symbolischer Name für die Control Engine kann geändert werden (siehe Abschnitt 9.5).

Einrichten einer Control Engine für lokalen Zugriff

Beindet sich die Control Engine auf dem gleichen Computer wie die Software SIMATIC Computing, müssen Sie die Control Engine in der Symboldatei so einrichten, daß sie auf den lokalen Computer zugreift:

Hinweis

Verwenden Sie diese Zeichen nicht im Feld für den Namen der Control Engine:
"/ , \ ' ' Diese Zeichen sind ungültig und werden von der Software nicht unterstützt.

1. Öffnen Sie die Symboldatei und wählen Sie die Control Engine. (Dem Abschnitt 9.3 entnehmen Sie Informationen zum Erstellen oder Öffnen einer Symboldatei.)
2. Wählen Sie den Menübefehl **Control Engine > Bearbeiten**, um das Dialogfeld "Konfiguration Control Engine" aufzurufen.
3. Tragen Sie, wie in Bild 9-8 gezeigt, **<lokal>** ein oder löschen Sie das Feld "Computername". Wenn Sie das Feld leer lassen, wird automatisch <lokal> eingefügt.
4. Wählen Sie die Schaltfläche "OK", um die Control Engine für lokalen Zugriff zu konfigurieren.

Wenn die Software SIMATIC Computing mit der Symboldatei auf die Control Engine zugreift, wird die Verbindung zur Control Engine auf dem lokalen Computer hergestellt.

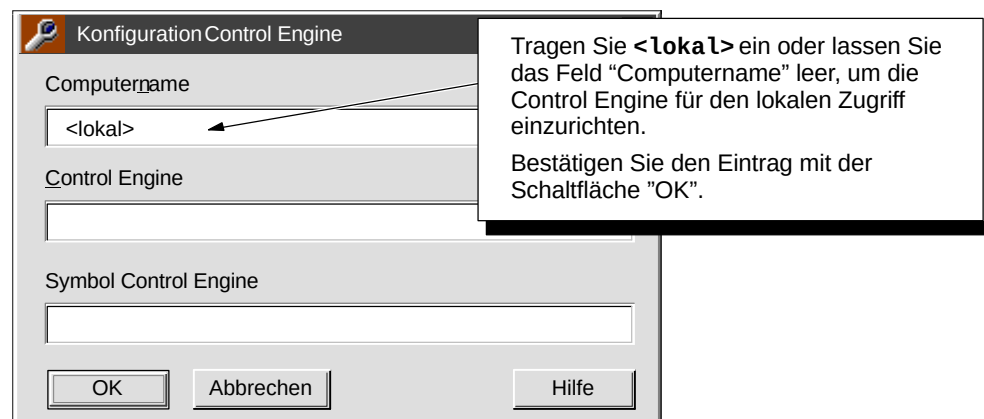


Bild 9-8 Einrichten einer Control Engine für lokalen Zugriff

Einrichten einer Control Engine für entfernten Zugriff

Beindet sich die Control Engine auf einem anderen Computer als die Software SIMATIC Computing, müssen Sie die Control Engine in der Symboldatei so einrichten, daß sie auf den entfernten Computer zugreift:

1. Öffnen Sie die Symboldatei und wählen Sie die Control Engine. (Dem Abschnitt 9.3 entnehmen Sie Informationen zum Erstellen oder Öffnen einer Symboldatei.)
2. Wählen Sie den Menübefehl **Control Engine > Bearbeiten**, um das Dialogfeld "Konfiguration Control Engine" aufzurufen.
3. Tragen Sie, wie in Bild 9-9 gezeigt, den Namen der entfernten Control Engine in das Feld "Computername" ein. Beispiel: Der Name der entfernten Control Engine in Bild 9-1 kann entweder **PC 2** oder **PC 3** sein.
4. Wählen Sie die Schaltfläche "OK", um die Control Engine für entfernten Zugriff zu konfigurieren.

Greift die Software SIMATIC Computing mit der Symboldatei auf die Control Engine zu, wird über das DCOM-Netz eine Verbindung zu der von Ihnen angegebenen Control Engine hergestellt.

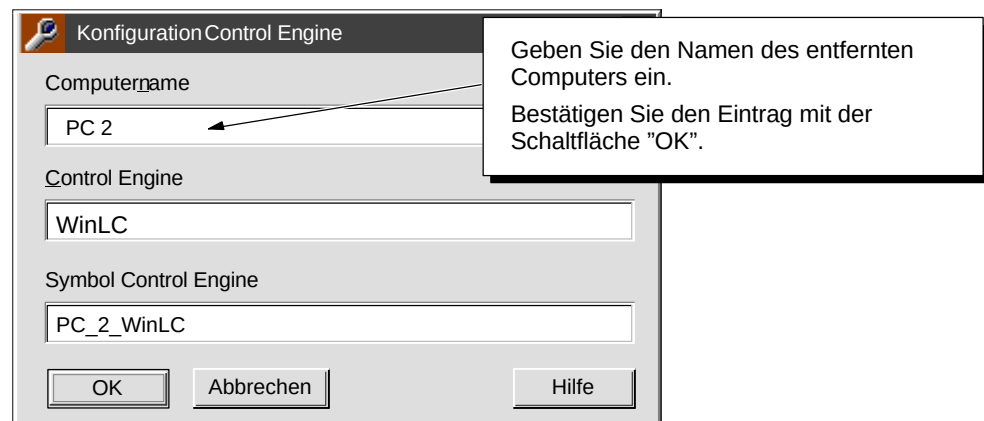


Bild 9-9 Einrichten einer Control Engine für entfernten Zugriff

9.5 Ändern des symbolischen Namens der Control Engine im Symboldatei-Editor

Sie können den symbolischen Namens der Control Engine im Symboldatei-Editor ändern:

1. Öffnen Sie die Symboldatei und wählen Sie die Control Engine. (Dem Abschnitt 9.3 entnehmen Sie Informationen zum Erstellen oder Öffnen einer Symboldatei.)
2. Wählen Sie den Menübefehl **Control Engine > Bearbeiten**, um das Dialogfeld "Konfiguration Control Engine" aufzurufen.
3. Tragen Sie, wie in Bild 9-10 gezeigt, den symbolischen Namen der Control Engine in das Feld "Symbol Control Engine" ein. Zum Beispiel kann der symbolische Name der in Bild 9-10 gezeigten Control Engine in PC2WinLC geändert werden.
4. Wählen Sie die Schaltfläche "OK", um den symbolischen Namen der Control Engine zu ändern.

Zum Umbenennen der Control Engine ändern Sie den Namen im Feld "Symbol Control Engine". Zum erneuten Zuweisen des Symbols ändern Sie den Namen im Feld "Symbol Control Engine".

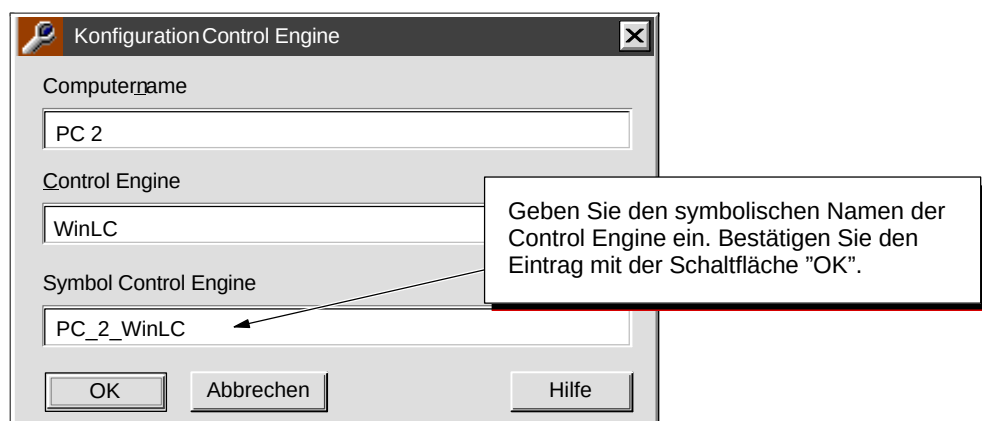


Bild 9-10 Einrichten einer Control Engine für entfernten Zugriff

Speicherbereiche von S7-Automatisierungssystemen



Übersicht

SIMATIC Computing gewährt Zugriff auf die Prozeßdaten in einer Control Engine, z.B. einem S7-Automatisierungssystem. Mit dem SIMATIC Data Control ermitteln Sie den Speicherbereich, auf den zugegriffen werden soll. Ausführliche Informationen zu Speicherbereichen entnehmen Sie dem Programmierhandbuch *Systemsoftware für S7-300/400 Programmwurf* und der Online-Hilfe in STEP 7.

Hinweis

In SIMATIC Computing können Sie nicht in Zeiten schreiben.

Kapitel	Beschreibung	Seite
A.1	Speicherbereiche von S7-Automatisierungssystemen	A-2
A.2	Zugreifen auf die S7-Datentypen	A-3
A.3	Beschreibungen der S7-Datentypen	A-6

A.1 Speicherbereiche von S7-Automatisierungssystemen

Tabelle A-1 führt die Speicherbereiche (einschließlich der internationalen und der SIMATIC Mnemonik) der S7-Automatisierungssysteme, auf die zugegriffen werden kann, auf. Beachten Sie folgende Regeln, wenn Sie auf die Peripheriebereiche der Ein- und Ausgänge (PE und PA) zugreifen:

- Auf die Speicherbereiche PE und PA kann nur im Byteformat zugegriffen werden (oder in größeren Einheiten, je nach Gerät). Sie können nicht auf einzelne Bits in diesen Bereichen zugreifen.
- Der Peripheriebereich der Eingänge (PE) überschreibt den Speicherbereich der Eingänge (E) zu Beginn eines jeden Zyklus. Wenn Sie mit einem ActiveX Control einen Wert im Speicherbereich der Eingänge, der über konfigurierte Peripherie verfügt, ändern, bleibt dieser Wert nur bis zum Beginn des nächsten Zyklus verändert, wenn der im Speicherbereich PE abgelegte Wert den geänderten Wert überschreibt.

Tabelle A-1 Speicherbereiche von S7-Automatisierungssystemen

Speicherbereich	Beschreibung
Peripheriebereich der Eingänge PI (International) PE (SIMATIC)	Dieser Speicherbereich überschreibt das Prozeßabbild der Eingänge zu Beginn eines jeden Zyklus. Sie können auf den Speicherbereich PE nur im Byteformat und nicht im Bitformat zugreifen.
Peripheriebereich der Ausgänge PQ (International) PA (SIMATIC)	Dieser Speicherbereich wird am Ende eines jeden Zyklus vom Prozeßabbild der Ausgänge überschrieben. Sie können auf den Speicherbereich PA nur im Byteformat und nicht im Bitformat zugreifen.
Prozeßabbild der Eingänge I (International) E (SIMATIC)	Dieser Speicherbereich wird vom Peripheriebereich der Eingänge zu Beginn eines jeden Zyklus überschrieben.
Prozeßabbild der Ausgänge Q (International) A (SIMATIC)	Dieser Speicherbereich überschreibt am Ende eines jeden Zyklus den Peripheriebereich der Ausgänge.
Merker M (International und SIMATIC)	Dieser Speicherbereich speichert Zwischenergebnisse, die vom Programm berechnet werden.
Timer T (International und SIMATIC)	Dieser Speicherbereich enthält die Zeiten, die vom Programm verwendet werden. In SIMATIC Computing können Sie Zeiten nur lesen. Sie können keine Daten in die Zeiten schreiben.
Counter C (International) Z (SIMATIC)	Dieser Speicherbereich enthält die Zähler, die vom Programm verwendet werden.
Datenbaustein DB (International und SIMATIC)	Die Adresse des DB verweist auf die Daten, die für das Programm in Datenbausteinen gespeichert sind.

A.2 Zugreifen auf die S7-Datentypen

Sie greifen auf Daten in der Control Engine zu, indem Sie einer Eigenschaft des Objekts eine Variable (eine Adresse im Speicher) der Control Engine zuordnen. Das höchstwertige Byte der Daten ist das Byte der Adresse (siehe Bild A-1). Das höchstwertige Byte für MW0 beispielsweise ist Byte 0 und das höchstwertige Byte für MD0 ist Byte 0. Tabelle A-2 führt die gültigen Datentypen auf, die Sie eingeben können, wenn Sie Variablen zuordnen.

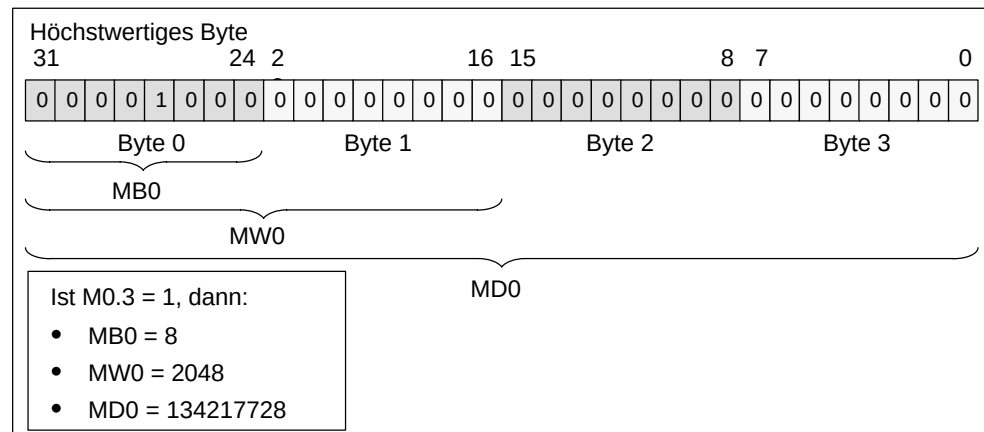


Bild A-1 Zugreifen auf Daten im Byte-, Wort- und Doppelwortformat

In SIMATIC Computing können Sie auch den Datentyp angeben, wenn Sie einer Eigenschaft eines Controls eine Variable zuordnen. Sie definieren den Datentyp, indem Sie die absolute Adresse im Speicher angeben, dann einen Doppelpunkt (:) gefolgt von dem Datentyp. Sie können beispielsweise für eine Variable den Datentyp REAL definieren, indem Sie "MD100:real" eingeben, wenn Sie die Variable zuordnen. Wenn Sie keinen Datentyp angeben, verwendet SIMATIC Computing die voreingestellten Datentypen, die in Tabelle A-2 aufgeführt sind.

Sie können auch auf Daten zugreifen, die in Feldern oder Zeichenketten gespeichert sind. Beispiel: Wenn Sie auf den zweiten Wert in einem eindimensionalen Feld mit Daten vom Typ REAL zugreifen möchten, geben Sie "MD100:real[2]" ein.

Hinweis

Sie können auf die meisten S7-Datentypen von anderen Anwendungen heraus zugreifen, ohne eine externe Auswertung der Daten liefern zu müssen. Diese Datentypen umfassen BOOL, BYTE, CHAR (Zeichen), WORD, DWORD, INT (Ganzzahl), DINT (Ganzzahl, 32 Bit) und REAL (Gleitpunkt). Einige der S7-Datentypen sind spezifisch für SIMATIC Produkte: DATE, S5TIME, TIME, TIME_OF_DAY (TOD) und DATE_AND_TIME.

Das SIMATIC Number Control wandelt diese Datentypen automatisch um. Wenn Sie jedoch auf die S7-spezifischen Datentypen mit anderen Controls zugreifen, müssen Sie die Daten auch manuell umwandeln. Die Beschreibungen zu diesen Datentypen folgen im Anschluß an Tabelle A-2.

Tabelle A-2 Adressieren von S7-Datentypen und S7-Speicherbereichen

Speicherbereich	Adresse	Gültiger Datentyp
Peripheriebereich der Ausgänge	PABx (SIMATIC) PQBx (International)	BYTE (Voreinstellung), CHAR
	PAWx (SIMATIC) PQWx (International)	WORD (Voreinstellung), INT, DATE, S5TIME
	PADx (SIMATIC) PQDx (International)	DWORD (Voreinstellung), DINT, REAL, TOD, TIME
Peripheriebereich der Eingänge	PEBx (SIMATIC) PIBx (International)	BYTE (Voreinstellung), CHAR
	PEWx (SIMATIC) PIWx (International)	WORD (Voreinstellung), INT, DATE, S5TIME
	PEDx (SIMATIC) PIDx (International)	DWORD (Voreinstellung), DINT, REAL, TOD, TIME
Ausgänge	Ax.y (SIMATIC) Qx.y (International)	BOOL (Voreinstellung)
	ABx (SIMATIC) QBx (International)	BYTE (Voreinstellung), CHAR
	AWx (SIMATIC) QWx (International)	WORD (Voreinstellung), INT, DATE, S5TIME
	ADx (SIMATIC) QDx (International)	DWORD (Voreinstellung), DINT, REAL, TOD, TIME
Eingänge	Ex.y (SIMATIC) Ix.y (International)	BOOL (Voreinstellung)
	EBx (SIMATIC) IBx (International)	BYTE (Voreinstellung), CHAR
	EWx (SIMATIC) IWx (International)	WORD (Voreinstellung), INT, DATE, S5TIME
	EDx (SIMATIC) IDx (International)	DWORD (Voreinstellung), DINT, REAL, TOD, TIME
Merker	Mx.y	BOOL (Voreinstellung)
	MBx	BYTE (Voreinstellung), CHAR
	MWx	WORD (Voreinstellung), INT, DATE, S5TIME
	MDx	DWORD (Voreinstellung), DINT, REAL, TOD, TIME
Datenbaustein (DB)	DBz.DBx.y DBz.DBXx.y	BOOL (Voreinstellung)
	DBz.DBBx	BYTE (Voreinstellung), CHAR
	DBz.DBWx	WORD (Voreinstellung), INT, DATE, S5TIME
	DBz.DBDx	DWORD (Voreinstellung), DINT, REAL, TOD, TIME

Tabelle A-2 Adressieren von S7-Datentypen und S7-Speicherbereichen, Fortsetzung

Speicherbereich	Adresse	Gültiger Datentyp
Timer (schreibgeschützt)	Tx	INT (Voreinstellung)
Counter	Zx (SIMATIC) Cx (International)	INT (Voreinstellung)

Hinweis

In SIMATIC Computing können Sie nicht in Zeiten schreiben.

Tabelle A-3 führt die S7-Datentypen und die entsprechenden Datentypen für C und Visual Basic auf.

Tabelle A-3 S7-Datentypen als Datentypen in C oder Visual Basic

S7-Datentyp	Datentyp in C	Datentyp in VB
ARRAY	VT_ARRAY	-/-
BOOL (Boolesch)	VT_BOOL	Boolean
BYTE	VT_UI1	Byte
CHAR (Zeichen)	VT_BSTR	String
DATE	VT_DATE	Date
DATE_AND_TIME	VT_DATE	Date
DINT (Ganzzahl, 32 Bit)	VT_I4	Long
DT (Datum und Zeit)	VT_DATE	Date
DWORD (Doppelwort)	VT_CY ¹	Currency ¹
INT (Ganzzahl)	VT_I2	Integer
REAL	VT_R4	Single
S5TIME	VT_I4	Long
STRING	VT_BSTR	String
TIME	VT_I4	Long
TOD (Tageszeit)	VT_DATE	Date
WORD	VT_I4	Long

¹ Wenn Sie Daten vom Datentyp DWORD in Excel einlesen, müssen Sie den Datentyp für die Excel-Datei bzw. für das Feld vom Datentyp "Allgemein" zum Datentyp "Zahl" umformatieren. Ansonsten formatiert Excel die S7-Daten vom Datentyp DWORD in Geldwerte, wobei in Excel der Datentyp "Currency" verwendet wird.



Vorsicht

Wenn Sie die Zeitfunktion oder Haltepunkte in Visual Basic mit SIMATIC Computing verwenden, kann dies zu Fehlern führen, die evtl. Ihren Computer bzw. die Anwendung zum Absturz bringen bzw. verriegeln können. Je nach Konfiguration kann die Anwendung dadurch die Kommunikation mit der Control Engine beenden. Dies kann unvorhersehbare Reaktionen im Prozeß auslösen, die zu Tod, schwerer Körperverletzung und/oder Sachschaden führen können.

Installieren Sie einen physikalischen NOT-AUS-Schaltkreis für die Maschine bzw. den Prozeß.

Timer in VB: Bei der Timer-Funktion in Visual Basic Version 5 kann ein Timer den in der gleichen Einweg-Bearbeitung befindlichen Code unterbrechen, wodurch Fehler mit schwerwiegenden Folgen auftreten können. Beachten Sie folgende Richtlinien, wenn Sie VB-Timer mit SIMATIC Computing einsetzen:

- Deaktivieren Sie die Timer immer im Unterprogramm Form_Unload. Andernfalls kann ein Timer ein Ereignis auslösen, während das VB-Programm geschlossen wird. Diese Bedingung kann einen Absturz bzw. eine Verriegelung Ihres Computers oder der Anwendung herbeiführen. Es kann auch sein, daß die Anwendung unsichtbar weiterläuft.
- Wenn Sie den Timer im Unterprogramm Form_Load starten, kann es sein, daß das Zeitereignis auftritt, bevor die anderen Objekte instanziiert wurden. Um sicherzustellen, daß die Objekte korrekt instanziiert werden, starten Sie einen Timer im Unterprogramm Form_Load immer mit einem großen Intervall (z.B. 1 oder 2 Sekunden). Nachfolgende Timer-Intervalle können kürzer sein.

A.3 Beschreibungen der S7-Datentypen

Zugreifen auf Daten in einem ARRAY

Jede Dimension eines Felds aus Bits, Bytes oder Zeichen ist auf Byte-Grenzen angeordnet. Bei allen anderen Feldern sind die Dimensionen auf Wort-Grenzen angeordnet. Bild A-2 zeigt ein Beispiel dafür, wie ein Feld im Speicher abgelegt wird. Das Betriebssystem berechnet die Bitadresse der Endposition eines jeden Elements im Feld. Das Feld wird dann bis zur Adresse des nächsten Worts (oder Bytes) belegt. Der nächste Datentyp beginnt auf der Grenze des nächsten Worts (bzw. Bytes).

Multi-dimensionale Felder werden sequentiell gespeichert. Im Beispiel in Bild A-2 folgt auf die Ganzzahl [1,1] die Ganzzahl [1,2] und auf die Ganzzahl [1,3] folgt die Ganzzahl [2,1].

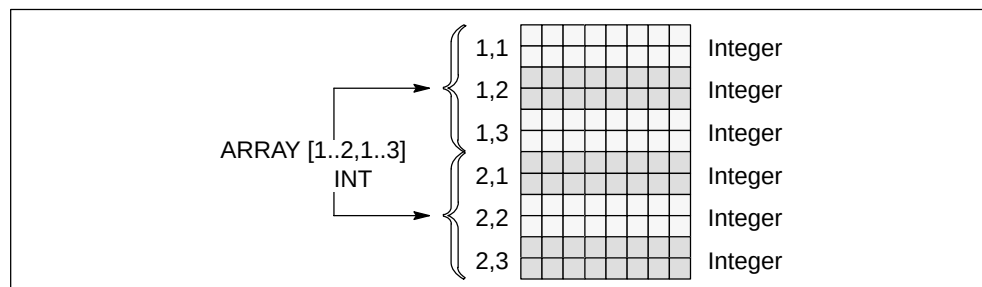


Bild A-2 Zugreifen auf Daten in einem ARRAY

Wenn Sie ein Feld aus der Control Engine lesen möchten, verwenden Sie die Eigenschaft `ReadVariable` des Data Control. Beispiel: Folgender Code liest ein eindimensionales Feld aus Ganzzahlen (mit Beginn an MW0) in das Feld "MyVariant".

```
S7Data1.ReadVariable("MW0:INT[100]", MyVariant, MyState, MyTimeout)
```

Folgender Code liest ein eindimensionales Feld aus 100 Bits (mit Beginn an M0.0) in das Feld "MyVariant":

```
S7Data1.ReadVariable("M0.0:[100]", MyVariant, MyState, MyTimeout)
```

In Abschnitt 5.11 finden Sie ein Beispielprogramm, das Felder liest und schreibt.

Zugreifen auf den Datentyp DATE

Der Datentyp DATE wird als positive Ganzzahl gespeichert und stellt die Anzahl Tage dar, die seit dem 1. Januar 1990 vergangen sind (siehe Bild A-3). Die gültigen Daten liegen zwischen dem 1. Januar 1990 und dem 31. Dezember 2168.

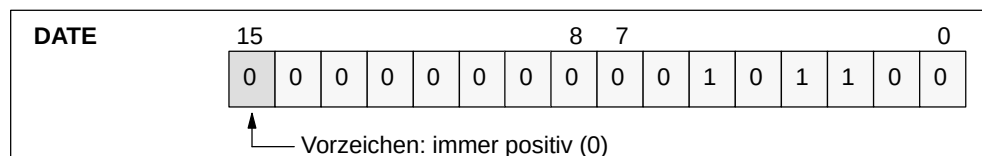


Bild A-3 Zugreifen auf den Datentyp DATE

Zugreifen auf den Datentyp DATE_AND_TIME

Der Datentyp DATE_AND_TIME wird als 8-Byte-Variante mit dem in Bild A-4 gezeigten Format gespeichert. Der Bereich für die einzelnen Bytes ist der folgende:

- Jahr: 1990 bis 1999, 2000 bis 2089 (BCD: 90h bis 99h, 90h bis 99h)
- Monat: 1 bis 12 (BCD: h bis 12h)
- Tag: 1 bis 31 (BCD: h bis 31h)
- Stunden: 00 bis 23 (BCD: 00h bis 23h)
- Minuten: 00 bis 59 (BCD: 00h bis 59h)
- Sekunden: 00 bis 59 (BCD: 00h bis 59h)
- Millisekunden: 0 bis 999 (BCD: 000h bis 999h)
- Wochentag: Sonntag (1) bis Samstag (7) (BCD: h bis 7h)

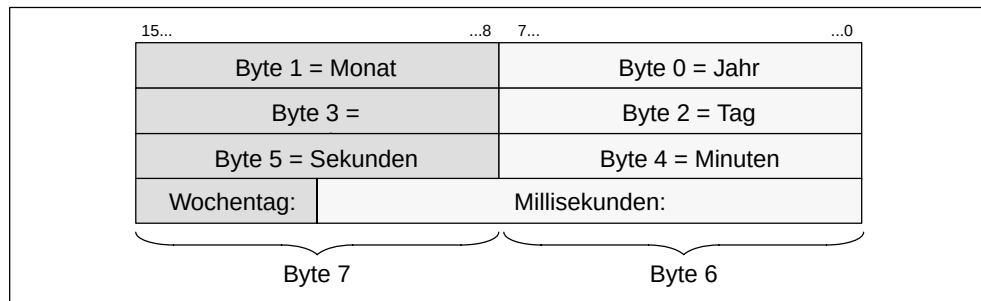


Bild A-4 Zugreifen auf den Datentyp DATE_AND_TIME

Zugreifen auf den Datentyp S5TIME

Im S7-Automatisierungssystem liefert der Datentyp S5TIME Angaben zur Zeitbasis und zum Zeitwert (im BCD-Format gespeichert). Wenn Sie jedoch mit SIMATIC Computing Daten vom Datentyp S5TIME lesen, gibt SIMATIC Computing eine Variable vom VB-Datentyp Long aus, die den Zeitwert in Millisekunden (ms) enthält.

Zugreifen auf Daten vom Datentyp STRING

Der Datentyp STRING ist eine Gruppierung (bzw. "Zeichenfolge") von ASCII-Zeichen und kann maximal 254 Zeichen enthalten. Jedes Zeichen in der Zeichenfolge wird in einem Byte gespeichert. Sie geben die Anzahl der Zeichen in der Zeichenfolge, auf die zugegriffen werden soll, an. Beispiel: STRING[25] greift auf eine Zeichenfolge mit 25 Zeichen zu.

Hinweis

Lesen und schreiben Sie Daten vom Datentyp STRING und CHAR als Visual Basic-Daten BSTR. Verwenden Sie kein Feld vom Datentyp CHAR, um den Datentyp STRING zu emulieren.

Verwenden Sie ein BSTR für jeden STRING bzw. jedes CHAR, unabhängig von der Länge der Daten, auf die zugegriffen wird. Beispiel:

- Zum Zugreifen auf CHAR[50] (das 50 Bytes bzw. 50 Zeichen angibt), verwenden Sie ein BSTR von bis zu 50 Bytes und **nicht** 50 einzelne BSTRs.
- Zum Zugreifen auf STRING[50] (das eine Zeichenfolge von 50 Zeichen angibt), verwenden Sie ein BSTR von bis zu 50 Bytes und **nicht** 50 einzelne BSTRs.

Wie Sie in Bild A-5 sehen, enthält der einer Zeichenfolge zugeordnet Speicher einen Header (2 Bytes), der folgende Angaben umfaßt:

- Das erste Byte speichert die maximale Speicherlänge für die Zeichenfolge (die Voreinstellung ist 256 Bytes).
- Das zweite Byte speichert den tatsächlichen Speicherumfang für die Zeichenfolge.

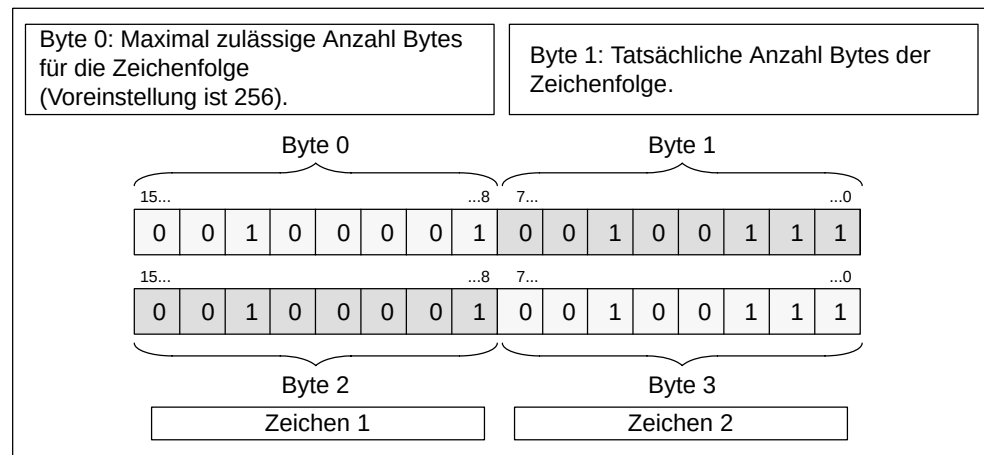


Bild A-5 Zugreifen auf Daten vom Datentyp STRING

Zugreifen auf den Datentyp TIME

Der Datentyp TIME wird gespeichert als Ganzzahl (32 Bit) mit Vorzeichen für die Anzahl Millisekunden (ms) im Bereich von -24 Tagen, 20 Stunden, 31 Minuten, 23 Sekunden und 648 ms bis +24 Tagen, 20 Stunden, 31 Minuten, 23 Sekunden und 647 ms.

Negative Werte werden als Zweierkomplement der positiven Zahl dargestellt. (Zum Bilden des Zweierkomplements der Daten invertieren Sie den Signalzustand aller Bits und addieren + 1 zum Ergebnis.)

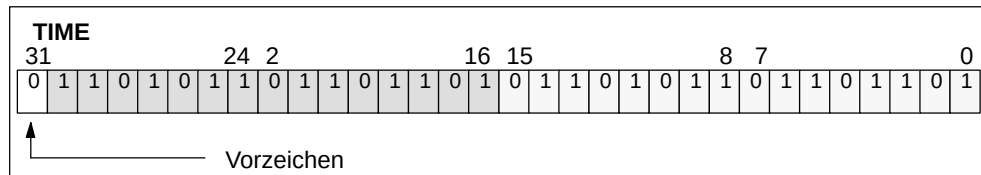


Bild A-6 Zugreifen auf den Datentyp TIME

Zugreifen auf den Datentyp TIME_OF_DAY (TOD)

Der Datentyp TIME_OF_DAY (TOD) wird gespeichert als positive Ganzzahl (32 Bit) der Anzahl an Millisekunden (ms), die seit Mitternacht vergangen sind, und zwar im Bereich zwischen 0:0:0.0 und 23:59:59.999.

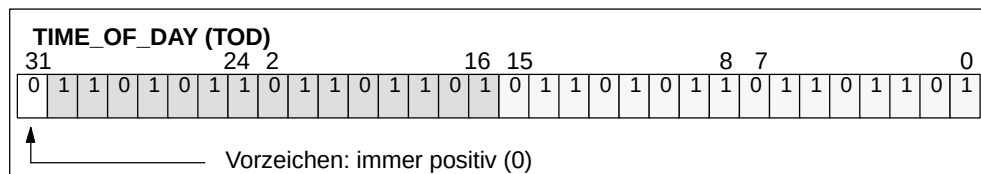


Bild A-7 Zugreifen auf den Datentyp TIME_OF_DAY (TOD)

Eigenschaften und Methoden

B.1 Methode **AboutBox**

Gültig für: Button Control, Edit Control, Label Control, Slider Control

Diese Methode zeigt das Dialogfeld "Info" des Control an.

Syntax:

object.**AboutBox**

Die Methode AboutBox besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.

B.2 Eigenschaft **Activated**

Gültig für: Data Control

Mit Hilfe dieser Eigenschaft können Sie angeben, ob alle Verbindungen aktiviert werden sollen oder nicht.

Syntax:

object.**Activated** [= *value*]

Die Eigenschaft Activated besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein boolescher Ausdruck, der angibt, ob das <i>Objekt</i> auf vom Anwender generierte Ereignisse reagieren kann.

Der *Wert* hat folgende Einstellungen:

Einstellung	Beschreibung
True	(Voreinstellung) Alle Verbindungen sind aktiviert.
False	Alle Verbindungen sind deaktiviert.

Hinweis

Die Verbindungen bleiben aufgebaut, auch wenn sie deaktiviert sind.

B.3 Eigenschaft Alignment

Gültig für: Button Control, Edit Control, Label Control

Mit Hilfe dieser Eigenschaft können Sie die Ausrichtung des Texts im Control angeben.

Syntax:

`object.Alignment [= value]`

Die Eigenschaft Alignment besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein Wert bzw. eine Konstante, der bzw. die die Ausrichtung festlegt.

Der *Wert* hat folgende Einstellungen:

Einstellung	Beschreibung
0 oder Left	(Voreinstellung für das Edit Control) Linksbündig.
1 oder Right	Rechtsbündig.
2 oder Center	(Voreinstellung für Button und Label Controls) Zentriert.

B.4 Eigenschaft Appearance

Gültig für: Button Control, Edit Control, Label Control

Ist für diese Eigenschaft die Option ThreeD (1) eingestellt und die Eigenschaft BorderStyle ist auf "Fixed Single" (1) gesetzt, dann erstellt die Eigenschaft Appearance Controls mit dreidimensionalem Aussehen. Ist für die Eigenschaft Flat (0) eingestellt, wird das Rechteck des Control von einem flachen Rand umgeben.

Hinweis

Diese Eigenschaft wirkt sich nur aus, wenn für die Eigenschaft BorderStyle die Option "Fixed Single" (1) eingestellt ist.

Syntax:

object.Appearance [= value]

Die Eigenschaft Appearance besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein Wert bzw. eine Konstante, der bzw. die das Aussehen des <i>Objekts</i> festlegt.

Der *Wert* hat folgende Einstellungen:

Einstellung	Beschreibung
0 oder Flat	Controls und andere Formen werden ohne plastische Wirkung dargestellt.
1 oder ThreeD	(Voreinstellung) Controls und andere Formen werden dreidimensional dargestellt.

B.5 Eigenschaft AutoConnect

Gültig für: Data Control

Mit Hilfe dieser Eigenschaft können Sie angeben, ob die konfigurierten Verbindungen zur Laufzeit aufgebaut werden oder nicht.

Syntax:

object.AutoConnect [= value]

Die Eigenschaft DefaultUpdateRate besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein boolescher Ausdruck, der angibt, ob das <i>Objekt</i> auf vom Anwender generierte Ereignisse reagieren kann.

Der Wert hat folgende Einstellungen:

Einstellung	Beschreibung
True	(Voreinstellung) Alle konfigurierten Verbindungen werden zur Laufzeit aufgebaut.
False	Die Verbindungen werden durch Aufruf der Methode Connect aufgebaut.

Hinweis

Wenn Sie in Ihrem Programm die Methode Connect explizit aufrufen, müssen Sie die Eigenschaft AutoConnect für das Data Control deaktivieren. So stellen Sie sicher, daß das Data Control nicht unerwarteterweise mit der Control Engine verbunden wird.

B.6 Eigenschaft AutoConnectTimeout

Gültig für: Data Control

Mit Hilfe dieser Eigenschaft können Sie ein Timeout angeben. Nach Ablauf der angegebenen Zeit gibt das Data Control einen Aufruf an die Methode Connect aus, sofern die Eigenschaft AutoConnect auf "True" gesetzt ist. Sie können den Wert auch im Register "General" im Dialogfeld "Eigenschaften" angeben.

Syntax:

***object.AutoConnectTimeout* [= *value*]**

Die Eigenschaft AutoConnectTimeout besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein Wert vom Datentyp "Long", der die Zeit für Timeout in Millisekunden angibt.

B.7 Eigenschaft BackColor

Gültig für: Edit Control, Label Control, Slider Control

Diese Eigenschaft stellt die Hintergrundfarbe für das Control ein.

Syntax:

object.BackColor [= *color*]

Die Eigenschaft BackColor besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>Color</i>	Ein Wert bzw. eine Konstante, der bzw. die die Farbe für den Hintergrund eines Objekts angibt.

Die *Farbe* hat folgende Einstellungen:

Einstellung	Beschreibung
Standardfarben	Farben, die durch die Verwendung der RGB-Farbpalette festgelegt sind.
Windows-Systemfarben	Farben, die von Systemkonstanten festgelegt sind (richten sich nach dem jeweiligen Container). In Visual Basic beispielsweise sind die Farben in der Objektbibliothek im Objekt-Browser aufgeführt.

B.8 Eigenschaft bDiagBuffOK

Gültig für: DBuffer Control

Diese schreibgeschützte Eigenschaft prüft, ob es eine Verbindung zum Diagnosepuffer der S7 Control Engine gibt.

Syntax:

object.bDiagBuffOK [= *value*]

Die Syntax der Eigenschaft bDiagBuffOK besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein boolescher Ausdruck, der angibt, ob es eine Verbindung zum Diagnosepuffer der S7 Control Engine gibt.

Der *Wert* hat folgende Einstellungen:

Einstellung	Beschreibung
True	Die Verbindung zum Diagnosepuffer wurde geprüft und ist aktiv.
False	(Voreinstellung) Es gibt keine Verbindung zum Diagnosepuffer.

B.9 Eigenschaft **bEngineConnected**

Gültig für: DBuffer Control

Diese schreibgeschützte Eigenschaft prüft, daß das Control eine Verbindung zu einer S7 Control Engine aufgebaut hat.

Syntax:

***object.bEngineConnected* [= *value*]**

Die Syntax der Eigenschaft **bEngineConnected** besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein boolescher Ausdruck, der angibt, ob das <i>Objekt</i> mit einer S7 Control Engine verbunden ist.

Der *Wert* hat folgende Einstellungen:

Einstellung	Beschreibung
True	Die Verbindung zur S7 Control Engine wurde geprüft und ist aktiv.
False	(Voreinstellung) Es gibt keine Verbindung zur S7 Control Engine.

B.10 Eigenschaft **BorderStyle**

Gültig für: Edit Control, Button Control, Label Control

Ist für die Eigenschaft der Wert "1-Fixed Single" eingestellt, dann ist das Control von einem rechteckigen Rahmen umgeben. Ist für die Eigenschaft der Wert "0-wNone" eingestellt, dann wird kein Rahmen angezeigt.

Hinweis

Diese Eigenschaft legt fest, ob die Eigenschaft Appearance wirksam ist.

Syntax:

***object.BorderStyle* [= *value*]**

Die Eigenschaft BorderStyle besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein Wert bzw. eine Konstante, der bzw. die den Rahmen des Control festlegt.

Der *Wert* hat folgende Einstellungen:

Einstellung	Beschreibung
0 oder None	(Voreinstellung) Kein Rahmen oder vom Rahmen abhängige Elemente.
1 oder FixedSingle	Fester, einfacher Rahmen.

B.11 Eigenschaft Caption

Gültig für: Label Control

Diese Eigenschaft gibt den Text an, der vom Control angezeigt werden soll.

Syntax:

***object.Caption* [= *value*]**

Die Eigenschaft Caption besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Eine Zeichenkette, die den Text für das Label angibt.

B.12 Methode Connect

Gültig für: Data Control

Diese Methode baut alle konfigurierten Verbindungen auf.

Hinweis

Wenn Ihr Unterprogramm programmatisch auf das Data Control zugreift, müssen Sie (mit der Methode Disconnect) im Unterprogramm Form_Unload immer die Verbindung zur Control Engine trennen.

Außerdem müssen Sie, wenn Sie die Methode Connect in Ihrem Programm explizit aufrufen, die Eigenschaft AutoConnect für das Data Control deaktivieren. So stellen Sie sicher, daß das Data Control nicht unerwarteterweise mit der Control Engine verbunden wird.

Syntax:

result = **object.Connect**

Die Methode Connect besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>result</i>	Ein Wert vom Datentyp "Long", der angibt, ob ein Fehler aufgetreten ist. Tritt kein Fehler auf, ist das Ergebnis Null.

B.13 Methode ConnectName

Gültig für: Data Control

Diese Methode baut für das im Formular angegebene Objekt Verbindungen auf.

Wenn Ihr Unterprogramm programmatisch auf das Data Control zugreift, müssen Sie (mit der Methode Disconnect) im Unterprogramm Form_Unload immer die Verbindung zur Control Engine trennen.

Außerdem müssen Sie, wenn Sie die Methode Connect in Ihrem Programm explizit aufrufen, die Eigenschaft AutoConnect für das Data Control deaktivieren. So stellen Sie sicher, daß das Data Control nicht unerwarteterweise mit der Control Engine verbunden wird.

Hinweis

Beim Programmieren mit Visual Basic (oder einer ähnlichen Programmiersprache) würde die Methode ConnectName eingesetzt werden, während beim Programmieren mit Visual C (oder einer ähnlichen Programmiersprache) die Methode ConnectObject verwendet werden würde.

Syntax:

result = **object.ConnectName** *ConnectedObject*, *ConnectionTable*

Die Methode `ConnectName` besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>result</i>	Ein Wert vom Datentyp "Long", der angibt, ob ein Fehler aufgetreten ist. Tritt kein Fehler auf, ist das Ergebnis Null.
<i>ConnectedObject</i>	Eine Zeichenkette, die den Namen des Objekts angibt, das verbunden werden soll. Enthält dieser Parameter keine Zeichenkette, dann erzeugt das Control das Ereignis <code>ValueChanged</code> , wenn sich eine verbundene Variable ändert.
<i>ConnectionTable</i>	(optional) Gibt eine Verbindungstabelle an. Entfällt der Parameter <code>ConnectionTable</code>, dann liest das Control die Eigenschaft <code>ConnectionTable</code> des verbundenen Objekts (<code>ConnectedObject</code>). Die Verbindungstabelle ist als Feld deklariert. Jedes Element im Feld besteht aus folgenden Teilen: • Name des Elements (z.B. "Value"). • Speicherplatz (z.B. MW100). • Aktualisierungsgeschwindigkeit bzw. Timeout-Wert (in ms). • Wert der Totzeit Ausführliche Informationen zur Verbindungstabelle entnehmen Sie dem Abschnitt 5.7.

Hinweis

Ist keiner der beiden Parameter `ConnectedObject` oder `ConnectionTable` angegeben, wird ein Fehler gemeldet.

B.14 Methode `ConnectObject`

Gültig für: Data Control

Diese Methode baut Verbindungen für ein bestimmtes Objekt auf, das im Programm deklariert wurde.

Wenn Ihr Unterprogramm programmatisch auf das Data Control zugreift, müssen Sie (mit der Methode `Disconnect`) im Unterprogramm `Form_Unload` immer die Verbindung zur Control Engine trennen.

Außerdem müssen Sie, wenn Sie die Methode `Connect` in Ihrem Programm explizit aufrufen, die Eigenschaft `AutoConnect` für das Data Control deaktivieren. So stellen Sie sicher, daß das Data Control nicht unerwarteterweise mit der Control Engine verbunden wird.

Hinweis

Beim Programmieren mit Visual Basic (oder einer ähnlichen Programmiersprache) würde die Methode `ConnectName` eingesetzt werden, während beim Programmieren mit Visual C (oder einer ähnlichen Programmiersprache) die Methode `ConnectObject` verwendet werden würde.

Syntax:

result = *object*.**ConnectObject** *ConnectedObject*, *ConnectionTable*

Die Methode `ConnectObject` besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>result</i>	Ein Wert vom Datentyp "Long", der angibt, ob ein Fehler aufgetreten ist. Tritt kein Fehler auf, ist das Ergebnis Null.
<i>ConnectedObject</i>	Eine Zeichenkette, die den Namen des Objekts angibt, das verbunden werden soll. Enthält dieser Parameter keine Zeichenkette, dann erzeugt das Control das Ereignis <code>ValueChanged</code> , wenn sich eine verbundene Variable ändert.
<i>ConnectionTable</i>	(optional) Gibt eine Verbindungstabelle an. Entfällt der Parameter <code>ConnectionTable</code> , dann liest das Control die Eigenschaft <code>ConnectionTable</code> des verbundenen Objekts (<code>ConnectedObject</code>). Die Verbindungstabelle ist als Feld deklariert. Jedes Element im Feld besteht aus folgenden Teilen: • Name des Elements (z.B. "Value"). • Speicherplatz (z.B. MW100). • Aktualisierungsgeschwindigkeit bzw. Timeout-Wert. • Wert der Totzeit

Hinweis

Ist keiner der beiden Parameter `ConnectedObject` oder `ConnectionTable` angegeben, wird ein Fehler gemeldet.

B.15 Eigenschaft **ControlEngine**

Gültig für: Data Control, DBuffer Control

Diese Eigenschaft speichert den Pfadnamen bzw. die Identifikation der Control Engine, die mit dem Control verbunden ist. Ausführliche Informationen zu den Zeichenfolgen der Control Engine entnehmen Sie dem Anhang G.

Syntax:

object.ControlEngine [= value]

Die Syntax der Eigenschaft ControlEngine besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Eine Zeichenkette, die den Pfadnamen bzw. die Identifikation der Control Engine angibt, auf die das "object" zugreifen soll.

B.16 Eigenschaft DataFormat

Gültig für: Edit Control

Diese Eigenschaft definiert die Speicherart, die für umgewandelte Werte verwendet wird. Wenn Sie ein Datenformat zum Anzeigen eines zu großen Werts verwenden, wird der Wert abgeschnitten.

Hinweis

Diese Eigenschaft legt fest, ob die Eigenschaft Precision wirksam ist.

Syntax:

object.DataFormat [= value]

Die Eigenschaft DataFormat besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein Wert bzw. eine Konstante, der bzw. die das Datenformat angibt (wie in Tabelle B-1 beschrieben).

Tabelle B-1 Einstellungen für das Datenformat

Konstante	Einstellung	Beschreibung
wBoolean	0	Bitwert
wBinary	1	Wert: Bit, Byte, Wort, Doppelwort, Ganzzahl (16 Bit) oder Ganzzahl (32 Bit).
wOctal	2	Wert: Byte, Wort, Doppelwort, Ganzzahl (16 Bit) oder Ganzzahl (32 Bit).
wHexadecimal	3	Wert: Byte, Wort, Doppelwort, Ganzzahl (16 Bit) oder Ganzzahl (32 Bit).
wUnsignedDecimal	4	Wert: Byte, Wort, Doppelwort, Ganzzahl (16 Bit) oder Ganzzahl (32 Bit).

Tabelle B-1 Einstellungen für das Datenformat, Fortsetzung

Konstante	Einstellung	Beschreibung
wSignedDecimal	5	Wert: Byte, Wort, Doppelwort, Ganzzahl (16 Bit) oder Ganzzahl (32 Bit).
wReal	6	Wert: vier Byte, Gleitpunkt
wTimer	7	Wert: mit Vorzeichen, zwei Byte, ganze Zahl
wCounter	8	Wert: mit Vorzeichen, zwei Byte, ganze Zahl
wTime	9	Wert: Ganzzahl mit Vorzeichen (IEC-Zeit)
wDate	10	Wert: Ganzzahl mit Vorzeichen (IEC-Datum)
wTimeOfDay	11	Wert: Ganzzahl mit Vorzeichen (IEC-Zeit)
wChar	12	ASCII-Zeichen, 1 Byte
wString	13	Zeichenfolge

Hinweis

Ist die konfigurierte Größe der Daten, auf die in der Control Engine zugegriffen werden soll, größer als die im Edit Control angezeigten Daten und ist der Wert der Daten aus dem Automatisierungssystem größer als vom Datenformat angezeigt werden kann, dann wird der Wert mit vorangestellten "..." angezeigt. Bevor der Wert im Edit Control geändert werden kann, müssen die vorangestellten "..." gelöscht werden.

Wird ein Wert aus dem Edit Control in das Automatisierungssystem mit dieser Konfiguration geschrieben, dann entspricht der Umfang der Daten, die in das Automatisierungssystem geschrieben werden, der Größe der Daten, die im Data Control konfiguriert ist. Deshalb müssen Sie vorsichtig vorgehen, damit nicht unbeabsichtigterweise Adressen im Speicher geändert werden.

B.17 Eigenschaft DefaultDeadband

Gültig für: Data Control

Mit Hilfe dieser Eigenschaft können Sie die Totzeit angeben, die das Data Control verwendet, wenn in der Verbindungstabelle keine Totzeit definiert ist.

Hinweis

Geben Sie eine Totzeit (z.B. "10") für eine Bitvariable (z.B. M15.5) an, überträgt die Control Engine den geänderten Wert für das Bit nicht.

Syntax:

`object.DefaultDeadBand [= value]`

Die Eigenschaft DefaultDeadBand besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein Wert vom Datentyp "Single", der nicht negativ sein darf.

B.18 Eigenschaft DefaultUpdateRate

Gültig für: Data Control

Mit Hilfe dieser Eigenschaft können Sie die Aktualisierungsgeschwindigkeit angeben, die das Data Control verwendet, wenn in der Verbindungstabelle keine Aktualisierungsgeschwindigkeit definiert ist.

Hinweis

In WinLC ist die voreingestellte Mindest-Aktualisierungsgeschwindigkeit 0. Für die CPU 416-2 DP ISA beträgt die voreingestellte Mindest-Aktualisierungsgeschwindigkeit 100 ms.

Syntax:

***object.DefaultUpdateRate* [= *value*]**

Die Eigenschaft DefaultUpdateRate besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein Wert vom Datentyp "Long".

Der Wert hat folgende Einstellungen:

Teil	Beschreibung
0	Alle Änderungen in der verbundenen Variable werden sofort gemeldet.
> 0	Änderungen in der verbundenen Variable werden nach diesem Timeout gemeldet.

B.19 Eigenschaft Direction

Gültig für: Slider Control

Diese Eigenschaft stellt die Ausrichtung (waagrecht oder senkrecht) des SIMATIC Control ein. Voreingestellt ist 0 - wHorizontal.

Syntax:

object.Direction [= *value*]

Die Eigenschaft Direction besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein Wert bzw. eine Konstante, der bzw. die die Ausrichtung festlegt.

Der *Wert* hat folgende Einstellungen:

Einstellung	Beschreibung
0	(Voreinstellung) wHorizontal
1	wVertical

B.20 Methode Disconnect

Gültig für: Data Control

Diese Methode trennt alle aufgebauten Verbindungen.

Hinweis

Wenn Ihr Unterprogramm programmatisch auf das Data Control zugreift, müssen Sie (mit der Methode Disconnect) im Unterprogramm Form_Unload immer die Verbindung zur Control Engine trennen.

Außerdem müssen Sie, wenn Sie die Methode Connect in Ihrem Programm explizit aufrufen, die Eigenschaft AutoConnect für das Data Control deaktivieren. So stellen Sie sicher, daß das Data Control nicht unerwarteterweise mit der Control Engine verbunden wird.

Syntax:

result = *object.Disconnect*

Die Methode Disconnect besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" (Kennzeichen eines bestimmten SIMATIC Control) bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>result</i>	Ein Wert vom Datentyp "Long", der angibt, ob ein Fehler aufgetreten ist. Tritt kein Fehler auf, ist das Ergebnis Null.

B.21 Eigenschaft DisplayFormatButtons

Gültig für: DBuffer Control

Mit Hilfe dieser Eigenschaft können Sie die Schaltflächen für das Format (Text oder Hex) anzeigen oder ausblenden.

Syntax:

***object*.DisplayFormatButtons [= value]**

Die Syntax der Eigenschaft DisplayFormatButtons besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein boolescher Ausdruck, der angibt, ob das <i>Objekt</i> die Schaltflächen anzeigen soll.

Der Wert hat folgende Einstellungen:

Einstellung	Beschreibung
True	(Voreinstellung) Die Schaltflächen "Text" und "Hex" werden angezeigt.
False	Die Schaltflächen "Text" und "Hex" werden nicht angezeigt.

B.22 Eigenschaft DisplayHelpButton

Gültig für: DBuffer Control

Mit Hilfe dieser Eigenschaft können Sie die Schaltfläche für die Online-Hilfe des Control anzeigen oder ausblenden.

Syntax:

***object*.DisplayHelpButton [= value]**

Die Syntax der Eigenschaft DisplayHelpButton besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein boolescher Ausdruck, der angibt, ob das <i>Objekt</i> die Schaltfläche anzeigen soll.

Der Wert hat folgende Einstellungen:

Einstellung	Beschreibung
True	(Voreinstellung) Die Schaltfläche für die Online-Hilfe des <i>Objekts</i> wird angezeigt.
False	Die Schaltfläche wird nicht angezeigt.

B.23 Eigenschaft DisplayHelpOnEventButton

Gültig für: DBuffer Control

Mit Hilfe dieser Eigenschaft können Sie die Schaltfläche für die Online-Hilfe zum markierten Diagnoseereignis anzeigen oder ausblenden.

Syntax:

***object*.DisplayHelpOnEventButton [= *value*]**

Die Syntax der Eigenschaft DisplayHelpOnEventButton besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein boolescher Ausdruck, der angibt, ob das <i>Objekt</i> die Schaltfläche anzeigen soll.

Der Wert hat folgende Einstellungen:

Einstellung	Beschreibung
True	(Voreinstellung) Die Schaltfläche für die Online-Hilfe zum markierten Diagnoseereignis wird vom <i>Objekt</i> angezeigt.
False	Die Schaltfläche wird nicht angezeigt.

B.24 Eigenschaft DisplayLowerPanel

Gültig für: DBuffer Control

Mit dieser Eigenschaft können Sie festlegen, ob das untere Teilfenster des Diagnosepuffers angezeigt oder ausgeblendet wird.

Syntax:

***object*.DisplayLowerPanel [= value]**

Die Syntax der Eigenschaft DisplayLowerPanel besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein boolescher Ausdruck, der angibt, ob das <i>Objekt</i> die Schaltfläche anzeigen soll.

Der Wert hat folgende Einstellungen:

Einstellung	Beschreibung
True	(Voreinstellung) Das untere Teilfenster des Diagnosepuffers wird vom <i>Objekt</i> angezeigt.
False	Das untere Teilfenster wird nicht angezeigt.

B.25 Eigenschaft DisplayUpdateButton

Gültig für: DBuffer Control

Diese Eigenschaft legt fest, ob die Schaltfläche zum Aktualisieren des Control - indem die Einträge des Diagnosepuffers aus der Control Engine gelesen werden - angezeigt oder ausgeblendet werden soll. Das Control liest den Diagnosepuffer nur, wenn eine Aktualisierung angefordert wird.

Syntax:

***object*.DisplayUpdateButton [= value]**

Die Syntax der Eigenschaft DisplayUpdateButton besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein boolescher Ausdruck, der angibt, ob das <i>Objekt</i> die Schaltfläche anzeigen soll.

Der *Wert* hat folgende Einstellungen:

Einstellung	Beschreibung
True	(Voreinstellung) Die Schaltfläche zum Aktualisieren des Diagnosepuffers wird vom <i>Objekt</i> angezeigt.
False	Die Schaltfläche wird nicht angezeigt.

B.26 Eigenschaft DisplayUpperPanel

Gültig für: DBuffer Control

Mit dieser Eigenschaft können Sie festlegen, ob das obere Teilfenster des Diagnosepuffers angezeigt oder ausgeblendet wird.

Syntax:

***object*.DisplayUpperPanel [= *value*]**

Die Syntax der Eigenschaft DisplayUpperPanel besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein boolescher Ausdruck, der angibt, ob das <i>Objekt</i> die Schaltfläche anzeigen soll.

Der *Wert* hat folgende Einstellungen:

Einstellung	Beschreibung
True	(Voreinstellung) Das obere Teilfenster des Diagnosepuffers wird <i>vom Objekt</i> angezeigt.
False	Das obere Teilfenster wird nicht angezeigt.

B.27 Eigenschaft DisplayValue

Gültig für: Edit Control, Slider Control

Diese Eigenschaft ist eine Variante, die den skalierten Wert für das Control ausgibt.

Syntax:

***object*.DisplayValue [= *value*]**

Die Eigenschaft DisplayValue besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Eine Variante, die den Wert des Control angibt.

B.28 Eigenschaft Enabled

Gültig für: Button Control, Edit Control, Label Control, Slider Control

Ist diese Eigenschaft auf True gesetzt, reagiert das Control auf Änderungen in der Eigenschaft Value und löst Ereignisse aus. Ist diese Eigenschaft auf False gesetzt, ist das Control deaktiviert und reagiert nicht auf Änderungen in der Eigenschaft Value und löst auch keine Ereignisse aus (mit Ausnahme des Fehlerereignisses).

Syntax:

***object.Enabled* [= *boolean*]**

Die Eigenschaft Enabled besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>Boolean</i>	Ein boolescher Ausdruck, der angibt, ob das <i>Objekt</i> auf vom Anwender generierte Ereignisse reagieren kann.

Die Einstellungen für *boolean* sind:

Einstellung	Beschreibung
True	(Voreinstellung) Das Objekt kann auf Ereignisse reagieren.
False	Das Objekt kann nicht auf Ereignisse reagieren.

B.29 Eigenschaft EnableSort

Gültig für: DBuffer Control

Mit dieser Eigenschaft können Sie festlegen, ob der Anwender die Einträge sortieren kann. Ist diese Eigenschaft aktiviert, kann der Anwender auf die Überschrift einer Spalte klicken und die Einträge werden entsprechend dieser Spalte sortiert.

Syntax:

object.EnableSort [= value]

Die Syntax der Eigenschaft EnableSort besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein boolescher Ausdruck, der die Sortierung der Spalten im oberen Teilfenster des Diagnosepuffers freigibt bzw. deaktiviert.

Der Wert hat folgende Einstellungen:

Einstellung	Beschreibung
True	(Voreinstellung) Die Einträge im oberen Teilfenster des Diagnosepuffers werden sortiert, wenn der Anwender auf den Kopf einer Spalte klickt.
False	Die Einträge können nicht sortiert werden.

B.30 Eigenschaft Factor

Gültig für: Edit Control, Slider Control

Die Eigenschaften Factor und Offset geben den Skalierfaktor und den Offset an, die benötigt werden, wenn Sie mit Hilfe einer Formel skalieren möchten.

Hinweis

Für die Eigenschaft ScaleMode muß "wByFormula" (1) eingestellt sein, damit die Eigenschaften Factor und Offset wirksam werden.

Sie können einen Wert mit Hilfe einer Formel skalieren. In der folgenden Formel ist der "Value" ähnlich dem Inhalt der Eigenschaft Value, wenn das Control mit einer Control Engine verbunden ist. "Factor" ist der Wert der Eigenschaft Factor und "Offset" ist der Wert der Eigenschaft Offset. "DisplayValue" ist der Inhalt der Eigenschaft Text.

$\text{Value} * \text{Factor} + \text{Offset} = \text{DisplayValue}$

Syntax:

object.Factor [= value]

Die Eigenschaft Factor besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein Gleitpunktwert, der den Faktor für die Formel zum Skalieren definiert.

Hinweis

Die Voreinstellung für Factor ist 1,0, die Voreinstellung für Offset ist 0,0.

B.31 Eigenschaft FalseCaption

Gültig für: Button Control

Diese Eigenschaft gibt den Text an, der im Control angezeigt wird, wenn die Eigenschaft "Value" auf "False" gesetzt ist (bzw. "0" oder "aus").

Syntax:

***object.FalseCaption* [= *string*]**

Die Eigenschaft FalseCaption besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>String</i>	Text, der den aktiven bzw. inaktiven Text des Control angibt.

B.32 Eigenschaft FalseColor

Gültig für: Button Control

Diese Eigenschaft gibt die Farbe des Control an, wenn die Eigenschaft "Value" auf "False" gesetzt ist (bzw. "0" oder "aus").

Syntax:

***object.FalseColor* [= *color*]**

Die Eigenschaft FalseColor besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>Color</i>	Ein Wert bzw. eine Konstante, der bzw. die die Farbe für den Vordergrund eines Objekts angibt.

Der Wert hat folgende Einstellungen:

Einstellung	Beschreibung
Standardfarben	Farben, die durch die Verwendung der RGB-Farbpalette festgelegt sind.
Windows-Systemfarben	Farben, die von Systemkonstanten festgelegt sind (richten sich nach dem jeweiligen Container). In Visual Basic beispielsweise sind die Farben in der Objektbibliothek im Objekt-Browser aufgeführt.

B.33 Eigenschaft FalsePicture

Gültig für: Button Control

Diese Eigenschaft gibt das deaktivierte (off, false usw.) Bild an, das im Control angezeigt wird.

Syntax:

***object.FalsePicture* [= *picture*]**

Die Eigenschaft FalsePicture besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>picture</i>	Ein Bild, das das Aussehen eines Controls festlegt.

B.34 Eigenschaft Font

Gültig für: Button Control, Edit Control, Label Control

Diese Eigenschaft gibt ein Objekt "Font" für die Hauptschriftart des Control aus.

Syntax:

***object.Font* [= *font*]**

Die Eigenschaft Font besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>Font</i>	Ein Wert, der die Schriftart für das Control angibt.

B.35 Eigenschaft ForeColor

Gültig für: Button Control, Edit Control, Label Control, Slider Control

Diese Eigenschaft stellt die Vordergrundfarbe ein, in der Text und Graphiken in einem Objekt angezeigt werden.

Syntax:

object.ForeColor [= *color*]

Die Eigenschaft ForeColor besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>Color</i>	Ein Wert bzw. eine Konstante, der bzw. die die Farbe für den Vordergrund eines <i>Objekts</i> angibt.

Die *Farbe* hat folgende Einstellungen:

Einstellung	Beschreibung
Standardfarben	Farben, die durch die Verwendung der RGB-Farbpalette festgelegt sind.
Windows-Systemfarben	Farben, die von Systemkonstanten festgelegt sind (richten sich nach dem jeweiligen Container). In Visual Basic beispielsweise sind die Farben in der Objektbibliothek im Objekt-Browser aufgeführt.

B.36 Eigenschaft FormatDisplay

Gültig für: DBuffer Control

Diese Eigenschaft dient dazu, das Format der zusätzlichen Informationen eines markierten Diagnoseeintrags zu ändern.

Syntax:

object.FormatDisplay [= *value*]

Die Syntax der Eigenschaft `FormatDisplay` besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein boolescher Ausdruck, der angibt, ob das <i>Objekt</i> die Informationen zum Diagnoseereignis als Text oder als Hexadezimalwerte anzeigt.

Der *Wert* hat folgende Einstellungen:

Einstellung	Beschreibung
True	(Voreinstellung) Die Informationen des Diagnoseereignisses werden als Text angezeigt.
False	Die Informationen werden als Hexadezimalzahlen angezeigt.

B.37 Eigenschaft `KnobHeight`

Gültig für: Slider Control

Diese Eigenschaft legt die Höhe der vom Control angezeigten Anzeige fest.

Syntax:

`object.KnobHeight [= single]`

Die Eigenschaft `KnobHeight` besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>Single</i>	Ein Wert, der die Höhe der Anzeige festlegt.

B.38 Eigenschaft `KnobPicture`

Gültig für: Slider Control

Diese Eigenschaft legt das Bild (die Graphik) für die Anzeige auf dem Control fest.

Syntax:

`object.KnobPicture [= picture]`

Die Eigenschaft KnobPicture besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>picture</i>	Ein Bild, das das Aussehen der Anzeige festlegt.

B.39 Eigenschaft KnobWidth

Gültig für: Slider Control

Diese Eigenschaft legt die Breite der vom Control angezeigten Anzeige fest.

Syntax:

***object.KnobWidth* [= *single*]**

Die Eigenschaft KnobWidth besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>Single</i>	Ein Wert, der die Breite der Anzeige festlegt.

B.40 Eigenschaft LargeChange

Gültig für: Slider Control

Diese Eigenschaft legt fest, wie weit sich die Bildlaufanzeige im Bildlaufeld bewegt, wenn das Control aktiviert ist und Sie eine der Tasten Seite-nach-oben/Seite-nach-unten drücken. Der Wert der Eigenschaft Value wird von der Eigenschaft LargeChange erhöht, wenn Sie die Taste Seite-nach-oben drücken oder rechts bzw. oberhalb der Bildlaufanzeige klicken. Der Wert wird von der Eigenschaft LargeChange verringert, wenn Sie die Taste Seite-nach-unten drücken oder links bzw. unterhalb der Bildlaufanzeige klicken.

Syntax:

***object.LargeChange* [= *value*]**

Die Eigenschaft LargeChange besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" (Kennzeichen eines bestimmten SIMATIC Control) bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein Wert bzw. eine Konstante, der bzw. die den Umfang der Änderungen angibt.

B.41 Eigenschaft Locked

Gültig für: Button Control, Edit Control, Slider Control

Ist das Control verriegelt, befindet es sich im schreibgeschützten Zustand. Sie können keine Werte ändern, doch der aktuelle Wert wird angezeigt. Standardmäßig ist das Control nicht verriegelt, so daß Sie Zahlen eingeben können.

Syntax:

object.Locked [= *boolean*]

Die Eigenschaft Locked besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>Boolean</i>	Ein boolescher Ausdruck, der angibt, ob das Control bearbeitet werden kann.

Die Einstellungen für *boolean* sind:

Einstellung	Beschreibung
True	Sie können im Text des Control blättern und Text markieren, Sie können den Text jedoch nicht bearbeiten. Änderungen der Eigenschaft Value werden wirksam. Das bedeutet, daß das Control die Werte des Automatisierungssystems anzeigt, aber daß Sie diese Werte nicht ändern können.
False	(Voreinstellung) Sie können den Text im Control bearbeiten.

B.42 Eigenschaften Max und Min

Gültig für: Edit Control, Slider Control

Ist für die Eigenschaft ScaleMode entweder wByRange oder wScaleNone eingestellt, geben diese Eigenschaften den maximalen/minimalen skalierten Wert des Control aus.

Syntax:

object.Max [= *value*]

object.Min [= *value*]

Die Eigenschaften Max und Min bestehen aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" (Kennzeichen eines bestimmten SIMATIC Control) bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein Wert bzw. eine Konstante, der bzw. die den maximalen/minimalen skalierten Wert des Control angibt.

B.43 Eigenschaft MultipleEngines

Gültig für: Data Control

Diese Eigenschaft gibt an, ob das Control mit einer oder mit mehreren Control Engines gleichzeitig verbunden ist. Ausführliche Informationen zu den Zeichenfolgen der Control Engine entnehmen Sie dem Anhang G.

Syntax:

object.MultipleEngines [= value]

Die Syntax der Eigenschaft MultipleEngines property besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein boolescher Ausdruck, der angibt, ob das <i>Objekt</i> mit einer oder mit mehreren Control Engines verbunden ist.

Der Wert hat folgende Einstellungen:

Einstellung	Beschreibung
True	Das <i>Objekt</i> ist mit mehreren Control Engines gleichzeitig verbunden.
False	(Voreinstellung) Das <i>Objekt</i> ist nur mit einer Control Engine verbunden, die von der Eigenschaft ControlEngine angegeben wird.

B.44 Eigenschaft Offset

Gültig für: Edit Control, Slider Control

Die Eigenschaften Factor und Offset geben den Skalierfaktor und den Offset an, die benötigt werden, wenn Sie mit Hilfe einer Formel skalieren möchten.

Hinweis

Für die Eigenschaft ScaleMode muß "wByFormula" (1) eingestellt sein, damit die Eigenschaften Factor und Offset wirksam werden.

Sie können einen Wert mit Hilfe einer Formel skalieren. In der folgenden Formel ist der "Value" ähnlich dem Inhalt der Eigenschaft Value, wenn das Control mit einer Control Engine verbunden ist. "Factor" ist der Wert der Eigenschaft Factor und "Offset" ist der Wert der Eigenschaft Offset. "DisplayValue" ist der Inhalt der Eigenschaft Text.

$\text{Value} * \text{Factor} + \text{Offset} = \text{DisplayValue}$

Syntax:

object.Offset [= *value*]

Die Eigenschaft Offset besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein Gleitpunktwert, der Factor bzw. Offset für die Formel zum Skalieren definiert.

Hinweis

Die Voreinstellung für Factor ist 1,0, die Voreinstellung für Offset ist 0,0.

B.45 Eigenschaft PCName

Gültig für: Data Control

Diese Eigenschaft gibt den Namen eines Computers (PC) an, damit die Control Engine über ein Netz, z.B. ein lokales Datennetz (LAN), verbunden werden kann. Ausführliche Informationen zu den Zeichenfolgen der Control Engine entnehmen Sie dem Anhang G.

Syntax:

object.PCName [= *value*]

Die Syntax der Eigenschaft PCName property besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Eine Zeichenkette, die den Pfadnamen bzw. die Identifikation des Computers (PC) für die Verbindung angibt.

B.46 Eigenschaft Picture

Gültig für: Slider Control, Label Control

Diese Eigenschaft legt das Bild (die Graphik) für das Control fest.

Syntax:

object.Picture [= *picture*]

Die Eigenschaft Picture besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>picture</i>	Ein Bild, das das Aussehen des Objekts festlegt.

B.47 Methode PopUpHelp

Gültig für: DBuffer Control

Diese Methode zeigt die Online-Hilfe des S7-Control an.

Syntax:

result = *object*.**PopUpHelp**

Die Syntax der Methode PropertyChangedObject besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>result</i>	Ein Wert vom Datentyp "Long", der angibt, ob ein Fehler aufgetreten ist. Tritt kein Fehler auf, ist das Ergebnis Null.

B.48 Methode PopUpHelpOnEvent

Gültig für: DBuffer Control

Diese Methode zeigt die Online-Hilfe zum markierten Diagnoseereignis an.

Syntax:

result = *object*.**PopUpHelpOnEvent**

Die Syntax der Methode PropertyChangedObject besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>result</i>	Ein Wert vom Datentyp "Long", der angibt, ob ein Fehler aufgetreten ist. Tritt kein Fehler auf, ist das Ergebnis Null.

B.49 Eigenschaft Precision

Gültig für: Edit Control

Diese Eigenschaft ist verfügbar, wenn die Eigenschaft DataFormat auf "Real" (6) gesetzt ist (Datentyp mit Genauigkeit). Dann können Sie die Genauigkeit für die Zahl ändern (Anzahl der Ziffern rechts vom Komma). Die Zahl wird an der angegebenen Nachkommastelle gerundet.

Hinweis

Sie müssen für die Eigenschaft DataFormat den Wert "Real" (6) einstellen, damit diese Eigenschaft wirksam ist.

Syntax:

object.Precision [= *value*]

Die Eigenschaft Precision besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein ganzzahliger Wert, der die Nachkommastellen der Zahl angibt. Voreingestellt ist der Wert 3.

B.50 Methode PropertyChangedName

Gültig für: Data Control

Diese Methode informiert das Data Control, daß sich der Wert einer Eigenschaft eines verbundenen Controls, das vom Objekt im Formular (z.B. S7Number1) angegeben wird, geändert hat. Das Data Control liest den Wert aus der Eigenschaft und schreibt es in die Datenquelle.

Hinweis

Beim Programmieren mit Visual Basic (oder einer ähnlichen Programmiersprache) würde die Methode PropertyChangedName eingesetzt werden, während beim Programmieren mit Visual C (oder einer ähnlichen Programmiersprache) die Methode PropertyChangedObject verwendet werden würde.

Syntax:

result = ***object.PropertyChangedName*** *ConnectedObject, Property*

Die Methode `PropertyChangedName` besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>result</i>	Ein Wert vom Datentyp "Long", der angibt, ob ein Fehler aufgetreten ist. Tritt kein Fehler auf, ist das Ergebnis Null.
<i>ConnectedObject</i>	Eine Zeichenkette, die den Namen eines verbundenen Controls angibt, dessen Eigenschaft sich geändert hat.
<i>Property</i>	Eine Zeichenkette, die den Namen der Eigenschaft angibt, die sich geändert hat.

B.51 Methode `PropertyChangedObject`

Gültig für: Data Control

Diese Methode informiert das Data Control, daß sich der Wert einer Eigenschaft eines verbundenen Controls (ein Objekt, das im Programm deklariert wird) geändert hat. Das Data Control liest den Wert aus der Eigenschaft und schreibt es in die Datenquelle.

Hinweis

Beim Programmieren mit Visual Basic (oder einer ähnlichen Programmiersprache) würde die Methode `PropertyChangedName` eingesetzt werden, während beim Programmieren mit Visual C (oder einer ähnlichen Programmiersprache) die Methode `PropertyChangedObject` verwendet werden würde.

Syntax:

result = *object*.**`PropertyChangedObject`** *ConnectedObject*, *Property*

Die Syntax der Methode `PropertyChangedObject` besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>result</i>	Ein Wert vom Datentyp "Long", der angibt, ob ein Fehler aufgetreten ist. Tritt kein Fehler auf, ist das Ergebnis Null.
<i>ConnectedObject</i>	Eine Zeichenkette, die den Namen eines verbundenen Controls angibt, dessen Eigenschaft sich geändert hat.
<i>Property</i>	Eine Zeichenkette, die den Namen der Eigenschaft angibt, die sich geändert hat.

B.52 Eigenschaft PushButton

Gültig für: Button Control

Diese Eigenschaft legt den Betriebszustand des Control fest: Ist PushButton = True oder 1, wird die Eigenschaft Value invertiert, solange das Button Control gedrückt ist (Ereignis MouseDown).

Syntax:

object.PushButton [= *boolean*]

Die Eigenschaft PushButton besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>Boolean</i>	Ein boolescher Ausdruck, der die Betriebsart des Control angibt.
Einstellung	Beschreibung
True	Die Schaltfläche ist gedrückt; die Eigenschaft Value wird invertiert.
False	(Voreinstellung) Die Schaltfläche ist nicht gedrückt.

B.53 Eigenschaften RawMax und RawMin

Gültig für: Edit Control, Slider Control

Diese Eigenschaften definieren die Bereiche zum Skalieren eines Werts:

- RawMax gibt den maximalen Rohwert des Control an, wenn die Eigenschaft ScaleMode auf wByRange oder wScaleNone gesetzt ist.
- RawMin gibt den minimalen Rohwert des Control an, wenn die Eigenschaft ScaleMode auf wByRange oder wScaleNone gesetzt ist.

Hinweis

Sie müssen für die Eigenschaft ScaleMode den Wert "wByRange" oder "wScaleNone" einstellen, damit diese Eigenschaften wirksam sind.

Wenn Sie durch Bereichsumwandlung skalieren möchten, geben Sie die unteren und oberen Grenzwerte des Quellbereichs (für den Wert in der Control Engine) und auch des Zielbereichs an (für den Wert der vom S7Number Control angezeigt werden soll). Die Werte des einen Bereichs werden in den anderen Bereich umgewandelt. Die Quell- und Zielbereiche definieren ein Verhältnis für die Umwandlung, sie geben keine oberen und unteren Grenzwerte an. Ein Wert kann größer oder kleiner als der Bereich sein. Während der Umwandlung werden die Werte mit Hilfe der zwei Bereiche extrapoliert.

Syntax:

object.**RawMax** [= *value*]

object.**RawMin** [= *value*]

Die Eigenschaften RawMax und RawMin bestehen aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein Wert bzw. eine Konstante, der bzw. die den maximalen oder minimalen Rohwert des Control angibt.

B.54 Methode ReadMultiVariables

Gültig für: Data Control

Diese Methode liest den Status der verbundenen Variablen in der Control Engine.

Syntax:

result = *object*.**ReadMultiVariables** (*VarNames*, *VarValues*, *States*)

Die Methode ReadMultiVariables besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>VarNames</i>	Eine Variante, die das Feld der Variablen (Adressen im Speicher) angibt, die aus der Control Engine gelesen werden sollen.
<i>VarValues</i>	Eine Variante, die das Feld der entsprechenden Werte der angegebenen Variablen aus der Control Engine enthält.
<i>States</i>	Eine Variante, die das Feld der Qualitätscodes (Long) für die Variablen enthält.
<i>result</i>	Ein Wert vom Datentyp "Long", der angibt, ob ein Fehler aufgetreten ist. Tritt kein Fehler auf, ist das Ergebnis Null.

B.55 Methode ReadVariable

Gültig für: Data Control

Diese Methode liest den Status einer bestimmten Variablen in der Control Engine.

Syntax:

result = *object*.**ReadVariable** (*VariableName*, *Value*, *State*, *TimeOut*)

Die Syntax der Methode ReadVariable besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>VariableName</i>	Eine Zeichenkette, die die Variable (Adresse im Speicher) in der Control Engine angibt, die gelesen werden soll.
<i>value</i>	Eine Variante, die den Inhalt der angegebenen Variablen der Control Engine enthält.
<i>State</i>	Ein Wert vom Datentyp "Long", der den Qualitätscode der Variablen angibt.
<i>TimeOut</i>	Ein Wert vom Datentyp "Long", der die Zeitdauer (in ms) vor Generieren eines Timeout-Fehlers angibt. (In dieser Release nicht implementiert.) In der aktuellen Release muß dieser Wert immer 0 sein.
<i>result</i>	Ein Wert vom Datentyp "Long", der angibt, ob ein Fehler aufgetreten ist. Tritt kein Fehler auf, ist das Ergebnis Null.

B.56 Eigenschaft ScaleMode

Gültig für: Edit Control, Slider Control

Diese Eigenschaft gibt an, auf welche Art Werte skaliert werden sollen. Sie können die Werte auch im Register "Scaling" im Dialogfeld "Eigenschaften" angeben. Sie können Werte auf drei Arten skalieren:

- Skalierung nach Formel (1-wByFormula):
$$\text{Value} * \text{Factor} + \text{Offset} = \text{DisplayValue}$$

Erklärung: Der "Value" ist ähnlich dem Inhalt der Eigenschaft Value, wenn das Control mit einer Control Engine verbunden ist. "Factor" ist der Wert der Eigenschaft Factor und "Offset" ist der Wert der Eigenschaft Offset. "DisplayValue" ist der Inhalt der Eigenschaft Text.
- Skalierung durch Bereichsumwandlung (2-wByRange): Sie geben einen Quellbereich (Werte aus der Control Engine) und einen Zielbereich (angezeigte Werte) an. Die Werte aus dem einen Bereich werden in den anderen Bereich umgewandelt.

Hinweis

Die Eigenschaft ScaleMode gibt an, ob die Eigenschaften RawMax, RawMin, Factor und Offset wirksam sind.

Syntax:

object.ScaleMode [= value]

Die Eigenschaft ScaleMode besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein Wert bzw. eine Konstante, der bzw. die die Art der Skalierung festlegt.

Der Wert hat folgende Einstellungen:

Einstellung	Beschreibung
wNoScaling (0)	(Voreinstellung) Keine Skalierung.
wByFormula (1)	Der Wert wird mit Hilfe der Formel, einschließlich "Factor" und "Offset", skaliert.
wByRange (2)	Der Wert wird durch Bereichsumwandlung skaliert.

B.57 Methode SelectEvent

Gültig für: DBuffer Control

Diese Methode markiert einen bestimmten Diagnoseeintrag im oberen Teilfenster des Control.

Syntax:

result = object.SelectEvent EventNumber

Die Syntax der Methode SelectEvent besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>result</i>	Ein Wert vom Datentyp "Long", der angibt, ob ein Fehler aufgetreten ist. Tritt kein Fehler auf, ist das Ergebnis Null.
<i>EventNumber</i>	Eine ganze Zahl, die die Nummer (Ereignis-ID) des Diagnoseereignisses darstellt.

B.58 Eigenschaft ShowErrorBoxes

Gültig für: Data Control

Diese Eigenschaft gibt an, ob die voreingestellten Fehlermeldungen angezeigt werden sollen, wenn ein vom Anwender generierter Fehler auftritt. Jedesmal, wenn ein Fehler auftritt, wird das Ereignis Error ausgelöst. Ist die Eigenschaft ShowErrorBoxes aktiviert, wird eine Standard-Fehlermeldung aufgeblendet.

Alle Fehler zu Verbindungen werden vom Ereignis ConnectionError gemeldet.

Hinweis

SIMATIC Computing verfügt nur über Fehlermeldungen in englischer Sprache. Wenn Sie Meldungen in anderen Sprachen anzeigen möchten, müssen Sie die Option ShowErrorBoxes deaktivieren und einen Programmcode für die Reaktion auf das Fehlerereignis schreiben.

Syntax:

object.ShowErrorBoxes [= *value*]

Die Eigenschaft ShowErrorBoxes besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein boolescher Ausdruck, der angibt, ob das Control Fehlermeldungen anzeigt.

Der Wert hat folgende Einstellungen:

Einstellung	Beschreibung
True	(Voreinstellung) Das Control zeigt Standard-Fehlermeldungen an.
False	Die Fehlermeldungen werden nicht angezeigt.

B.59 Eigenschaft ShowMinMax

Gültig für: Slider Control

Diese Eigenschaft gibt an, ob das Control den Wertebereich (Minimum und Maximum) anzeigt.

Syntax:

object.ShowMinMax [= *boolean*]

Die Eigenschaft ShowMinMax besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>Boolean</i>	Ein boolescher Ausdruck, der angibt, ob das Control den Wertebereich anzeigt.

Die Einstellungen für *boolean* sind:

Einstellung	Beschreibung
True	(Voreinstellung) Das Control zeigt den Mindest- und Höchstwert an.
False	Das Control zeigt den Wertebereich nicht an.

B.60 Eigenschaft SmallChange

Gültig für: Slider Control

Diese Eigenschaft legt fest, wie weit sich die Anzeige im Bildlauf Feld bewegt, wenn das Control aktiviert ist und Sie eine der Richtungstasten nach oben/unten oder nach rechts/links drücken. Die Eigenschaft Value wird durch SmallChange erhöht, wenn Sie eine der Richtungstasten nach rechts oder oben drücken. Sie wird durch SmallChange verringert, wenn Sie eine der Richtungstasten nach links oder unten drücken.

Syntax:

object.SmallChange [= *value*]

Die Eigenschaft SmallChange besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein Wert bzw. eine Konstante, der bzw. die den Umfang der Änderungen angibt.

B.61 Eigenschaft StretchMode

Gültig für: Button Control, Slider Control, Label Control

Diese Eigenschaft gibt den Stretch-Modus (centered, resize image, resize frame, smart tile oder tile) des Control an. Diese Eigenschaft wirkt sich nur aus, wenn für die Eigenschaft Style der Wert 1 - wGraphical eingestellt ist.

Syntax:

object.StretchMode [= *value*]

Die Eigenschaft StretchMode besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Kennung des spezifischen Slider Control.
<i>value</i>	Eine Konstante, die den Stretch-Modus festlegt (wie unter "Einstellungen" beschrieben).

Der Wert hat folgende Einstellungen:

Einstellung	Beschreibung
0	wCentered Das Bitmap wird im Control zentriert.
1	wResizelImage (Voreinstellung) Die Größe des Bitmap wird angepaßt (vergrößert oder verkleinert), damit es ins Control paßt.
2	wResizeFrame Die Größe des Control-Rahmens wird der Größe des Bitmap angepaßt.
3	wSmartTile Das Bitmap wird durch Duplizieren von benachbarten Rechtecken vergrößert, damit es ins Control paßt. Diese Einstellung funktioniert am besten bei einfarbigen Bitmaps mit Rahmen.
4	wTile Das Bitmap wird, sofern es kleiner ist als das Control, dupliziert und unter- und nebeneinander dargestellt, um das Control auszufüllen.

B.62 Eigenschaft Style

Gültig für: Button Control, Slider Control, Label Control

Diese Eigenschaft stellt die Darstellungsart (standard oder graphisch) des Control ein.

Syntax:

object.Style [= value]

Die Eigenschaft Style besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Kenntnis des spezifischen Slider Control.
<i>value</i>	Eine Konstante, die die Darstellungsart festlegt (wie unter "Einstellungen" beschrieben).

Der Wert hat folgende Einstellungen:

Einstellung	Beschreibung
0	WStandard (verwendet interne Zeichenmethoden)
1	wGraphical (Voreinstellung) (verwendet Bitmaps)

B.63 Eigenschaft TagSource

Gültig für: Data Control

Diese Eigenschaft definiert die Quelle der symbolischen Informationen, die verwendet werden soll, wenn Variablen zugeordnet und Control Engines gekennzeichnet werden sollen. Bei der Quelle kann es sich um eine Symboldatei handeln. Ausführliche Informationen zu den Zeichenfolgen der Control Engine entnehmen Sie dem Anhang G.

Syntax:

object.TagSource [= value]

Die Syntax der Eigenschaft TagSource besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Eine Zeichenkette, die den Pfad der Quelle (z.B. einer Symboldatei) für symbolische Informationen angibt, die zum Konfigurieren des Control für Variablen und Control Engines verwendet werden sollen.

B.64 Eigenschaft Text

Gültig für: Edit Control

Diese Eigenschaft gibt den Text an, der vom Control angezeigt werden soll.

Syntax:

object.Text [= *value*]

Die Eigenschaft Text besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Eine Zeichenfolge, die den Text angibt, der auf dem Control angezeigt werden soll.

B.65 Eigenschaft Ticks

Gültig für: Slider Control

Diese Eigenschaft stellt die Anzahl der Teilstriche für das Control ein. Haben Sie beispielsweise Ticks = 10 eingestellt, wird die Skala des Control in 10 Abschnitte unterteilt.

Syntax:

object.Ticks [= *value*]

Die Eigenschaft Ticks besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein Wert bzw. eine Konstante, der bzw. die die Anzahl der anzuzeigenden Teilstriche angibt.

B.66 Eigenschaft TrueCaption

Gültig für: Button Control

Diese Eigenschaft gibt den Text an, der im Control angezeigt wird, wenn die Eigenschaft "Value" auf "True" gesetzt ist (bzw. "1" oder "ein").

Syntax:

object.TrueCaption [= *string*]

Die Eigenschaft TrueCaption besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>String</i>	Text, der den aktiven bzw. inaktiven Text des Control angibt.

B.67 Eigenschaft TrueColor

Gültig für: Button Control

Diese Eigenschaft gibt die Farbe des Control an, wenn die Eigenschaft "Value" auf "True" gesetzt ist (bzw. "1" oder "ein").

Syntax:

object.TrueColor [= *color*]

Die Eigenschaft TrueColor besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>Color</i>	Ein Wert bzw. eine Konstante, der bzw. die die aktive bzw. inaktive Farbe eines Controls angibt (wie unter "Einstellungen" beschrieben).

Der Wert hat folgende Einstellungen:

Einstellung	Beschreibung
Standardfarben	Farben, die durch die Verwendung der RGB-Farbpalette festgelegt sind.
Windows-Systemfarben	Farben, die von Systemkonstanten festgelegt sind (richten sich nach dem jeweiligen Container). In Visual Basic beispielsweise sind die Farben in der Objektbibliothek im Objekt-Browser aufgeführt.

B.68 Eigenschaft TruePicture

Gültig für: Button Control

Diese Eigenschaft gibt das aktivierte (on, true usw.) Bild an, das im Control angezeigt wird.

Syntax:

object.**TruePicture** [= *picture*]

Die Eigenschaft TruePicture besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>picture</i>	Ein Bild, das das Aussehen eines Controls festlegt.

B.69 Methode Update

Gültig für: DBuffer Control

Diese Methode liest den Diagnosepuffer der Control Engine und aktualisiert die Informationen, die im Control angezeigt werden.

result = *object*.**Update**

Syntax:

Die Syntax der Methode Update besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>result</i>	Ein Wert vom Datentyp "Long", der angibt, ob ein Fehler aufgetreten ist. Tritt kein Fehler auf, ist das Ergebnis Null.

B.70 Eigenschaft Value

Gültig für: Button Control, Edit Control, Slider Control

Diese Eigenschaft muß über das Data Control mit einem Wert im Automatisierungssystem verbunden werden. Sie ist verknüpfbar.

Edit Control - Die Eigenschaft Value ist eine Variante, die den (nicht skalierten) Wert des Control ausgibt.

Button Control - Diese Eigenschaft Value zeigt den Zustand der Schaltfläche an.

Slider Control - Diese Eigenschaft zeigt die Position der Bildlaufanzeige für das Slider Control an.

Hinweis

Ändert sich der Wert der Eigenschaft Value, wird das Ereignis Change erzeugt.

Syntax:

***object.Value* [= value]**

Die Eigenschaft Value besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Eine Variante, die den Wert des Control angibt.

B.71 Eigenschaft WriteMode

Gültig für: Edit Control

Diese Eigenschaft legt die Reaktion des Control fest, wenn der Anwender einen neuen Wert eingibt. Ist die Eigenschaft WriteMode auf Automatic (0) gesetzt, wird der Wert (sofern er gültig ist) automatisch in die Eigenschaft Value (und in die Control Engine) geschrieben. Ist die Eigenschaft WriteMode auf Manual (1) gesetzt, dann wird der eingegebene Wert erst dann in die Eigenschaft Value geschrieben, wenn Ihr Programmcode die Methode "Write" für das Control aufruft.

Syntax:

***object.WriteMode* [= value]**

Die Eigenschaft WriteMode besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein Wert bzw. eine Konstante, der bzw. die angibt, ob das Control automatisch die eingegebenen Werte an die Eigenschaft Value weitergeben soll:

Der Wert hat folgende Einstellungen:

Einstellung	Beschreibung
Automatic (0)	(Voreinstellung) Neue Eingabewerte werden automatisch an die Eigenschaft Value weitergegeben.
Manual (1)	Neue Eingabewerte werden erst dann an die Eigenschaft Value weitergegeben, wenn das Control die Methode "Write" bearbeitet.

B.72 Methode WriteNow

Gültig für: Edit Control

Diese Methode gibt einen geänderten Wert für die Eigenschaft Value des Control aus. Dies muß nur dann manuell ausgeführt werden, wenn die Eigenschaft WriteMode auf Manual (1) gesetzt ist.

Syntax:

***object*.WriteNow**

Die Methode WriteNow besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.

B.73 Methode WriteMultiVariables

Gültig für: Data Control

Diese Methode schreibt neue Werte in verschiedene Variablen in der Control Engine.

Syntax:

***result* = *object*.WriteMultiVariables (VarNames, VarValues, States)**

Die Methode WriteMultiVariables besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>VarNames</i>	Eine Variante, die das Feld der Variablen (Adressen im Speicher) in der Control Engine angibt.
<i>VarValues</i>	Eine Variante, die das Feld der entsprechenden Werte, die in die angegebenen Variablen geschrieben werden sollen, enthält.
<i>States</i>	Eine Variante, die das Feld der Qualitätscodes (Long) für die Variablen enthält.
<i>result</i>	Ein Wert vom Datentyp "Long", der angibt, ob ein Fehler aufgetreten ist. Tritt kein Fehler auf, ist das Ergebnis Null.

B.74 Methode WriteVariable

Gültig für: Data Control

Diese Methode schreibt einen neuen Wert in eine bestimmte Variable in der Control Engine.

Syntax:

result = *object*.**WriteVariable** (*VariableName*, *Value*, *TimeOut*)

Die Syntax der Methode WriteVariable besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>VariableName</i>	Eine Zeichenkette, die die Variable (Adresse im Speicher) in der Control Engine angibt.
<i>value</i>	Eine Variante, die den Inhalt enthält, der in die angegebene Variable in der Control Engine geschrieben werden soll.
<i>TimeOut</i>	Ein Wert vom Datentyp "Long", der die Zeitdauer (in ms) vor Generieren eines Timeout-Fehlers angibt. (In dieser Release nicht implementiert.) In der aktuellen Release muß dieser Wert immer 0 sein.
<i>result</i>	Ein Wert vom Datentyp "Long", der angibt, ob ein Fehler aufgetreten ist. Tritt kein Fehler auf, ist das Ergebnis Null.

B.75 Eigenschaft ZeroPad

Gültig für: Edit Control

Diese Eigenschaft gibt an, ob die im Control angezeigte Zahl bis zur Größe des Datentyps mit Nullen aufgefüllt wird (links vom Wert).

Syntax:

object.ZeroPad [= *value*]

Die Eigenschaft ZeroPad besteht aus folgenden Teilen:

Teil	Beschreibung
<i>object</i>	Der Ausdruck "object" (Kennzeichen eines bestimmten SIMATIC Control) bezieht sich auf eines der unter "Gültig für" aufgeführten Objekte.
<i>value</i>	Ein boolescher Ausdruck, der angibt, ob den angezeigten Zahlen führende Nullen vorangestellt werden:

Der Wert hat folgende Einstellungen:

Teil	Beschreibung
True	Der Zahl werden so viele führende Nullen vorangestellt, bis die von der Eigenschaft DataType angegebene Größe erreicht ist.
False	(Voreinstellung) Der Zahl werden keine führenden Nullen vorangestellt.

Events

C.1 Ereignis Change

Gültig für: Button Control, Edit Control, Label Control, Slider Control

Dieses Ereignis tritt auf, wenn sich der Wert der Eigenschaft Value ändert. Der Wert in der Eigenschaft Value kann entweder von der Control Engine oder vom S7 Control geändert werden.

Syntax: **Change()**

C.2 Ereignis Click

Gültig für: Button Control, Edit Control, Label Control, DBuffer Control, Slider Control

Dieses Ereignis tritt auf, wenn sich der Mauszeiger auf dem Control befindet und eine Maustaste gedrückt und wieder losgelassen wird.

Syntax: **Click()**

Hinweis

Zum Unterscheiden zwischen linker, rechter und mittlerer Maustaste setzen Sie die Ereignisse MouseDown und MouseUp ein. Enthält das Ereignis Click Code, wird das Ereignis DblClick nie ausgelöst, weil das Ereignis Click als erstes der beiden Ereignisse ausgelöst wird. Daraus resultiert, daß das Ereignis Click dem Mausklick zuvorkommt, so daß das Ereignis DblClick nicht auftritt.

C.3 Ereignis ConnectionError

Gültig für: Data Control

Dieses Ereignis tritt auf, wenn in einer Verbindung ein Fehler auftritt.

Syntax:

ConnectionError(*State As Long, ConnectedObject As Object, _
Property As String, Variable As String*)

Das Ereignis `ConnectionError` besteht aus folgenden Teilen:

Teil	Beschreibung
<i>State</i>	Ein Wert vom Datentyp "Long" mit dem Zustand der Verbindung.
<i>ConnectedObject</i>	Ein Objektausdruck, der das verbundene Objekt angibt.
<i>Property</i>	Eine Zeichenkette mit dem Namen der Eigenschaft.
<i>Variable</i>	Eine Zeichenkette mit dem Namen der verbundenen Variablen.

C.4 Ereignis `DblClick`

Gültig für: Edit Control, Label Control, Slider Control

Dieses Ereignis tritt auf, wenn sich der Mauszeiger auf dem Control befindet und mit einer Maustaste doppelt geklickt wird.

Syntax: **`DblClick()`**

Hinweis

Zum Unterscheiden zwischen linker, rechter und mittlerer Maustaste setzen Sie die Ereignisse `MouseDown` und `MouseUp` ein.

Enthält das Ereignis `Click` Code, wird das Ereignis `DblClick` nie ausgelöst, weil das Ereignis `Click` als erstes der beiden Ereignisse ausgelöst wird. Daraus resultiert, daß das Ereignis `Click` dem Mausklick zuvorkommt, so daß das Ereignis `DblClick` nicht auftritt.

C.5 Ereignis `Error`

Gültig für: Button Control, Edit Control, Label Control, Slider Control

Dieses Ereignis tritt auf, wenn das Control einen Fehler erkennt.

Syntax:

**`Error(long SCode, BSTR lpszDescription, BSTR lpszHelpFileName, _
long nHelpId)`**

Das Ereignis `Error` besteht aus folgenden Teilen:

Teil	Beschreibung
<i>SCode</i>	(siehe Tabelle C-1)
<i>lpszDescription</i>	Zeichenkette, die eine Beschreibung des Fehlercodes enthält.

Teil	Beschreibung
<i>lpSzHelpFileName</i>	Name der Hilfedatei, in der der Fehler beschrieben ist.
<i>nHelpId</i>	Kennzeichen des Hilfethemas, das eine Beschreibung des Fehlers enthält.

Tabelle C-1 SCodes (Fehlercodes)

Name	Value	Beschreibung
wFACTOR_ZERO	0xC0040002	Factor: Darf nicht Null sein.
wRAWMINMAX	0xC0040006	RawMin muß kleiner sein als RawMax.
wMINMAX	0xC0040009	Min muß kleiner sein als Max.
wLARGECHANGE_ZERO	0xC004000A	Large Change: Muß größer sein als Null und kleiner als ...
wTICKS_ZERO_100	0xC004000C	Ticks: Muß sich um eine Zahl zwischen 1 und 100 handeln.
wKNOBHEIGHT_ZERO	0xC004000E	Knob Height: Muß größer sein als Null.
wKNOBWIDTH_ZERO	0xC0040010	Knob Width: Muß größer sein als Null.
wSMALLCHANGE_ZERO	0xC0040012	Small Change: Muß größer sein als Null und kleiner als
wRAWMIN_SCALEMODE	0xC0040014	RawMin darf nur gesetzt werden, wenn für ScaleMode wByRange eingestellt ist.
wRAWMAX_SCALEMODE	0xC0040015	RawMax darf nur gesetzt werden, wenn für ScaleMode wByRange eingestellt ist.
wEDIT_OUT_OF_RANGE	0xC0040016	Wert liegt außerhalb des Bereichs.
wEDIT_WRONGVALUE	0xC0040017	Es wurde ein falscher Wert eingestellt.
wBIGFONT	0xC0040018	Warning: Schriftgröße ist zu groß.
wPREC_RANGE	0xC004001A	Precision: Muß sich um eine Zahl zwischen 0 und 7 handeln.

C.6 Ereignis KeyDown

Gültig für: Button Control, Edit Control, Slider Control

Dieses Ereignis tritt auf, wenn Sie eine Taste drücken, während ein Control aktiviert bzw. im Fokus ist. Siehe auch: Ereignis KeyUp.

Syntax: **KeyDown(long KeyID, long Shift)**

Das Ereignis KeyDown besteht auf folgenden Teilen:

Teil	Beschreibung
<i>KeyID</i>	Code für Tasten wie vbKeyF1 (Taste F1) oder vbKeyHome (Taste Pos1). Wenn Sie Code für Tasten angeben möchten, verwenden Sie die Konstanten aus der Objektbibliothek im Objekt-Browser in Visual Basic (VB).
<i>Shift</i>	Eine ganze Zahl, die dem Zustand der Tasten SHIFT, CTRL und ALT bei Auftreten des Ereignisses entspricht. Bei dem Parameter Shift handelt es sich um ein Bitfeld, das die niederwertigsten Bits enthält, die den Tasten SHIFT (Bit 0), CTRL (Bit 1) und ALT (Bit 2) entsprechen. Diese Bits entsprechen jeweils den Werten 1, 2 und 4. Einige, alle oder keines der Bits kann gesetzt sein, wodurch angezeigt wird, daß einige, alle oder keine der Tasten gedrückt wurden. Beispiel: Sind die Tasten CTRL und ALT beide gedrückt, ist der Wert von Shift 6.

Setzen Sie die Vorgehensweisen für die Ereignisse KeyDown und KeyUp ein, wenn Sie sowohl auf das Drücken als auch auf das Loslassen einer Taste reagieren müssen.

KeyDown und KeyUp werten die Klein- und Großschreibung der Zeichen durch zwei Parameter aus: Der Parameter KeyID zeigt die physikalische Taste an (gibt deshalb "A" und "a" als eine Taste aus). Der Parameter Shift zeigt den Zustand für Shift+Taste an und gibt deshalb entweder "A" oder "a" aus.

Müssen Sie den Parameter Shift prüfen, können Sie die Shift-Konstanten einsetzen, die die Bits des Parameters definieren. Die Konstanten haben folgende Werte:

- vbShiftMask (1): Bitmaske der Taste SHIFT
- vbCtrlMask (2): Bitmaske der Taste CTRL
- vbAltMask (4): Bitmaske der Taste ALT

Die Konstanten dienen als Bitmasken, mit denen Sie beliebige Tastenkombinationen prüfen können.

Sie prüfen einen Zustand, indem Sie zunächst jedes Ergebnis einer temporären ganzzahligen Variablen zuordnen und anschließend den Parameter "Shift" mit einer Bitmaske vergleichen. Durch eine UND-Verknüpfung mit dem Parameter "Shift" können Sie prüfen, ob der Zustand größer als 0 ist, wodurch angezeigt wird, daß die Taste SHIFT gedrückt wurde.

C.7 Ereignis KeyPress

Gültig für: Button Control, Edit Control, Slider Control

Dieses Ereignis tritt auf, wenn eine ANSI-Taste gedrückt und wieder losgelassen wird, während das Control aktiviert bzw. im Fokus ist.

Syntax: **KeyPress (long keyAscii)**

Das Ereignis KeyPress besteht auf folgenden Teilen:

Teil	Beschreibung
<i>keyAscii</i>	ASCII ist ein Code für die gedrückten Tasten wie vbKeyF1 (Taste F1) oder vbKeyHome (Taste Pos1).

C.8 Ereignis KeyUp

Gültig für: Button Control, Edit Control, Slider Control

Dieses Ereignis tritt auf, wenn eine Taste gedrückt und wieder losgelassen wird, während das Control aktiviert bzw. im Fokus ist.

Syntax: **KeyUp(long KeyID, long Shift)**

Das Ereignis KeyDown besteht auf folgenden Teilen:

Teil	Beschreibung
<i>KeyID</i>	Code für Tasten wie vbKeyF1 (Taste F1) oder vbKeyHome (Taste Pos1). Wenn Sie Code für Tasten angeben möchten, verwenden Sie die Konstanten aus der Objektbibliothek im Objekt-Browser in Visual Basic (VB).
<i>Shift</i>	Eine ganze Zahl, die dem Zustand der Tasten SHIFT, CTRL und ALT bei Auftreten des Ereignisses entspricht. Bei dem Parameter Shift handelt es sich um ein Bitfeld, das die niederwertigsten Bits enthält, die den Tasten SHIFT (Bit 0), CTRL (Bit 1) und ALT (Bit 2) entsprechen. Diese Bits entsprechen jeweils den Werten 1, 2 und 4. Einige, alle oder keines der Bits kann gesetzt sein, wodurch angezeigt wird, daß einige, alle oder keine der Tasten gedrückt wurden. Beispiel: Sind die Tasten CTRL und ALT beide gedrückt, ist der Wert von Shift 6.

Setzen Sie die Vorgehensweisen für die Ereignisse KeyDown und KeyUp ein, wenn Sie sowohl auf das Drücken als auch auf das Loslassen einer Taste reagieren müssen.

KeyDown undKeyUp werten die Klein- und Großschreibung der Zeichen durch zwei Parameter aus: Der Parameter KeyID zeigt die physikalische Taste an (gibt deshalb "A" und "a" als eine Taste aus). Der Parameter Shift zeigt den Zustand für Shift+Taste an und gibt deshalb entweder "A" oder "a" aus.

Müssen Sie den Parameter Shift prüfen, können Sie die Shift-Konstanten einsetzen, die die Bits des Parameters definieren. Die Konstanten haben folgende Werte:

- vbShiftMask (1): Bitmaske der Taste SHIFT
- vbCtrlMask (2): Bitmaske der Taste CTRL
- vbAltMask (4): Bitmaske der Taste ALT

Die Konstanten dienen als Bitmasken, mit denen Sie beliebige Tastenkombinationen prüfen können.

Sie prüfen einen Zustand, indem Sie zunächst jedes Ergebnis einer temporären ganzzahligen Variablen zuordnen und anschließend den Parameter "Shift" mit einer Bitmaske vergleichen. Durch eine UND-Verknüpfung mit dem Parameter "Shift" können Sie prüfen, ob der Zustand größer als 0 ist, wodurch angezeigt wird, daß die Taste SHIFT gedrückt wurde.

C.9 Ereignis MouseDown

Gültig für: Button Control, Edit Control, Label Control, Slider Control

Dieses Ereignis tritt auf, wenn sich der Mauszeiger auf dem Control befindet und die Maustaste gedrückt wird.

Syntax:

```
MouseDown(short Button, short Shift, OLE_XPOS_PIXELS x, _  
OLE_YPOS_PIXELS y)
```

Das Ereignis `MouseDown` besteht auf folgenden Teilen:

Teil	Beschreibung
<i>Button</i>	Eine ganze Zahl, die die Maustaste angibt, die gedrückt wurde und das Auftreten des Ereignisses verursacht hat. Der Parameter "Button" ist ein Bitfeld mit Bits, die der linken Maustaste (Bit 0), der rechten Maustaste (Bit 1) und der mittleren Maustaste (Bit 2) entsprechen. Diese Bits entsprechen jeweils den Werten 1, 2 und 4. Nur eines der Bits ist gesetzt und zeigt dadurch an, welche Maustaste das Ereignis ausgelöst hat.
<i>Shift</i>	Eine ganze Zahl, die dem Zustand der Tasten SHIFT, CTRL und ALT beim Drücken bzw. Loslassen der Maustaste, die im Parameter Button angegeben ist, entspricht. Ein Bit wird gesetzt, wenn die Taste gedrückt ist. Bei dem Parameter Shift handelt es sich um ein Bitfeld, das die niederwertigsten Bits enthält, die den Tasten SHIFT (Bit 0), CTRL (Bit 1) und ALT (Bit 2) entsprechen. Diese Bits entsprechen jeweils den Werten 1, 2 und 4. Der Parameter "Shift" zeigt den Zustand dieser Tasten an. Einige, alle oder keines der Bits kann gesetzt sein, wodurch angezeigt wird, daß einige, alle oder keine der Tasten gedrückt ist. Beispiel: Sind die Tasten CTRL und ALT beide gedrückt, ist der Wert von Shift 6.
<i>x,y</i>	Gibt eine Zahl aus, die der aktuellen Position des Mauszeigers entspricht.

C.10 Ereignis `MouseMove`

Gültig für: Button Control, Edit Control, Label Control, Slider Control

Dieses Ereignis tritt auf, wenn Sie den Mauszeiger über ein Control bewegen.

Syntax:

```
MouseMove(short Button, short Shift, OLE_XPOS_PIXELS x, _  
OLE_YPOS_PIXELS y)
```

Das Ereignis `MouseMove` besteht auf folgenden Teilen:

Teil	Beschreibung
<i>Button</i>	Eine ganze Zahl, die die Maustaste angibt, die gedrückt wurde und das Auftreten des Ereignisses verursacht hat. Der Parameter "Button" ist ein Bitfeld mit Bits, die der linken Maustaste (Bit 0), der rechten Maustaste (Bit 1) und der mittleren Maustaste (Bit 2) entsprechen. Diese Bits entsprechen jeweils den Werten 1, 2 und 4. Nur eines der Bits ist gesetzt und zeigt dadurch an, welche Maustaste das Ereignis ausgelöst hat.
<i>Shift</i>	Eine ganze Zahl, die dem Zustand der Tasten SHIFT, CTRL und ALT beim Drücken bzw. Loslassen der Maustaste, die im Parameter Button angegeben ist, entspricht. Ein Bit wird gesetzt, wenn die Taste gedrückt ist. Bei dem Parameter Shift handelt es sich um ein Bitfeld, das die niederwertigsten Bits enthält, die den Tasten SHIFT (Bit 0), CTRL (Bit 1) und ALT (Bit 2) entsprechen. Diese Bits entsprechen jeweils den Werten 1, 2 und 4. Der Parameter "Shift" zeigt den Zustand dieser Tasten an. Einige, alle oder keines der Bits kann gesetzt sein, wodurch angezeigt wird, daß einige, alle oder keine der Tasten gedrückt ist. Beispiel: Sind die Tasten CTRL und ALT beide gedrückt, ist der Wert von Shift 6.
<i>x,y</i>	Gibt eine Zahl aus, die der aktuellen Position des Mauszeigers entspricht.

C.11 Ereignis `MouseUp`

Gültig für: Button Control, Edit Control, Label Control, Slider Control

Dieses Ereignis tritt auf, wenn eine Maustaste losgelassen wird, während sich der Mauszeiger über dem Control befindet.

Syntax:

```
MouseUp(short Button, short Shift, OLE_XPOS_PIXELS x, _  
OLE_YPOS_PIXELS y)
```

Das Ereignis MouseUp besteht auf folgenden Teilen:

Teil	Beschreibung
<i>Button</i>	Eine ganze Zahl, die die Maustaste angibt, die gedrückt wurde und das Auftreten des Ereignisses verursacht hat. Der Parameter "Button" ist ein Bitfeld mit Bits, die der linken Maustaste (Bit 0), der rechten Maustaste (Bit 1) und der mittleren Maustaste (Bit 2) entsprechen. Diese Bits entsprechen jeweils den Werten 1, 2 und 4. Nur eines der Bits ist gesetzt und zeigt dadurch an, welche Maustaste das Ereignis ausgelöst hat.
<i>Shift</i>	Eine ganze Zahl, die dem Zustand der Tasten SHIFT, CTRL und ALT beim Drücken bzw. Loslassen der Maustaste, die im Parameter Button angegeben ist, entspricht. Ein Bit wird gesetzt, wenn die Taste gedrückt ist. Bei dem Parameter Shift handelt es sich um ein Bitfeld, das die niederwertigsten Bits enthält, die den Tasten SHIFT (Bit 0), CTRL (Bit 1) und ALT (Bit 2) entsprechen. Diese Bits entsprechen jeweils den Werten 1, 2 und 4. Der Parameter "Shift" zeigt den Zustand dieser Tasten an. Einige, alle oder keines der Bits kann gesetzt sein, wodurch angezeigt wird, daß einige, alle oder keine der Tasten gedrückt ist. Beispiel: Sind die Tasten CTRL und ALT beide gedrückt, ist der Wert von Shift 6.
<i>x,y</i>	Gibt eine Zahl aus, die der aktuellen Position des Mauszeigers entspricht.

C.12 Ereignis ValueChanged

Gültig für: Data Control

Dieses Ereignis tritt auf, wenn sich der Wert einer verbundenen Variablen ändert und für den Aufruf der Methode Connect kein verbundenes Ereignis angegeben war. Das Ereignis ValueChanged können Sie auch im Register "Events" im Dialogfeld "Eigenschaften" konfigurieren.

Syntax:

ValueChanged(*Property As String, Variable As String, Value as _ Variant, Quality as Integer*)

Das Ereignis ValueChanged besteht auf folgenden Teilen:

Teil	Beschreibung
<i>Property</i>	Eine Zeichenkette mit dem Namen der Eigenschaft.
<i>Variable</i>	Eine Zeichenkette mit dem Namen der verbundenen Variablen.
<i>value</i>	Eine Variante mit dem neuen Wert der Variablen.
<i>Quality</i>	Gibt eine ganze Zahl mit der Qualität des neuen Werts aus.

Einrichten der SIMATIC Computing Konfiguration

D

Kapitelübersicht

Mit dem Werkzeug SIMATIC Computing-Konfiguration können Sie die Kommunikation an eine Control Engine richten. Sie können mit dem Werkzeug auch die Sprache für die Software SIMATIC Computing einstellen, den OPC-Server einrichten und Anwendungen, die mit früheren Versionen von SIMATIC Computing erstellt wurden und keine Symboldateien unterstützen, verbinden.

Hinweis

Es darf immer nur eine Control Engine aktiv sein. Anstatt das Data Control in Ihrem Programm neu zu konfigurieren, können Sie mit dem Konfigurationswerkzeug eine andere Control Engine für die Verbindung einstellen.

Kapitel	Beschreibung	Seite
D.1	Konfigurieren der OPC-Verbindung	D-2
D.2	Auswählen der Sprache	D-5
D.3	Auswählen der Control Engine für ältere Programme	D-6
D.4	Einrichten der Kommunikation in der Anwendung "PG/PC-Schnittstelle einstellen"	D-7

D.1 Konfigurieren der OPC-Verbindung

In SIMATIC Computing können Sie mit OPC die Verbindung zu einer oder zu mehreren Control Engines herstellen. Sie können die Verbindung zur Control Engine auch über ein Netz herstellen, z.B. ein lokales Datennetz (LAN).

Wie Sie in Bild D-1 sehen, bietet die Anwendung zum Einrichten von OPC in WinAC folgende Optionen, um eine Verbindung zu einer Control Engine aufzubauen.

- Über die Option "Verbindung über Symbolquelle" können Sie eine Verbindung zu mehreren Control Engines gleichzeitig herstellen. Sie geben die Symboldatei an, die die Symbole für die Variablen und Control Engines, auf die zugegriffen werden soll, angibt.
- Mit der Option "Direkte Verbindung" können Sie die Verbindung zu einer bestimmten Control Engine auf einem bestimmten Computer herstellen. Bei einer direkten Verbindung können Sie keine Symboldatei verknüpfen und auch keine Symbole verwenden, um auf Daten in der Control Engine zuzugreifen. Bei dieser Option geben Sie den Namen der gewünschten Control Engine sowie den Namen des Computers, auf der sich die Control Engine befindet, an.

Geben Sie folgende Zeichenketten im Feld "Control Engine" ein, um die unterschiedlichen SIMATIC Automatisierungssysteme anzugeben:

- **WinLC** (WinLC von WinAC Basis)
- **wcS7=3** (für eine Slot-PLC wie die CPU 416-2 DP ISA von WinAC Pro)
- **wcS7=xx, a, b** (für andere SIMATIC Automatisierungssysteme im MPI-Netz an, Erläuterung: xx ist die MPI-Adresse, a ist die Nummer des Baugruppenträgers und b ist die Nummer des Steckplatzes.)
- **wcIP=xxx.xxx.xxx.xxx,a,b** für eine Control Engine in einem TCP/IP LAN oder **wcMAC=xx.xx.xx.xx.xx.xx,a,b** für eine Control Engine in einem Industrial Ethernet mit STEP 7 V5 SP3

Ausführliche Informationen zu den Zeichenfolgen der Control Engine entnehmen Sie dem Anhang G.

Konfigurieren der OPC-Verbindung

Gehen Sie folgendermaßen vor, um die OPC-Verbindungen im WinAC OPC-Setup einzurichten:

1. Wählen Sie im Windows Startmenü den Befehl **SIMATIC > PC Based Control > SIMATIC Computing-Konfiguration**, um das Dialogfeld "SIMATIC Computing-Konfiguration" aufzurufen. Öffnen Sie das Register "OPC" (siehe Bild D-1).
2. Wenn Sie die Verbindung zu einer bestimmten Control Engine einrichten möchten, ohne Symbole zu verwenden, wählen Sie die Option "Direkte Verbindung" und tragen den Namen des gewünschten Computers und den Namen der gewünschten Control Engine ein.

3. Wenn Sie zum Zugreifen auf Daten in der Control Engine Symbole verwenden möchten oder wenn Sie auf mehrere Control Engines zugreifen möchten, wählen Sie die Option "Verbindung über Symbolquelle" und tragen den Namen der Symboldatei ein. (Wählen Sie die Schaltfläche "Blättern", um nach einer Symboldatei zu suchen.)
4. Wählen Sie die Schaltfläche "Übernehmen", um die Daten zu übernehmen und bestätigen Sie mit "OK", um das Dialogfeld zu schließen.

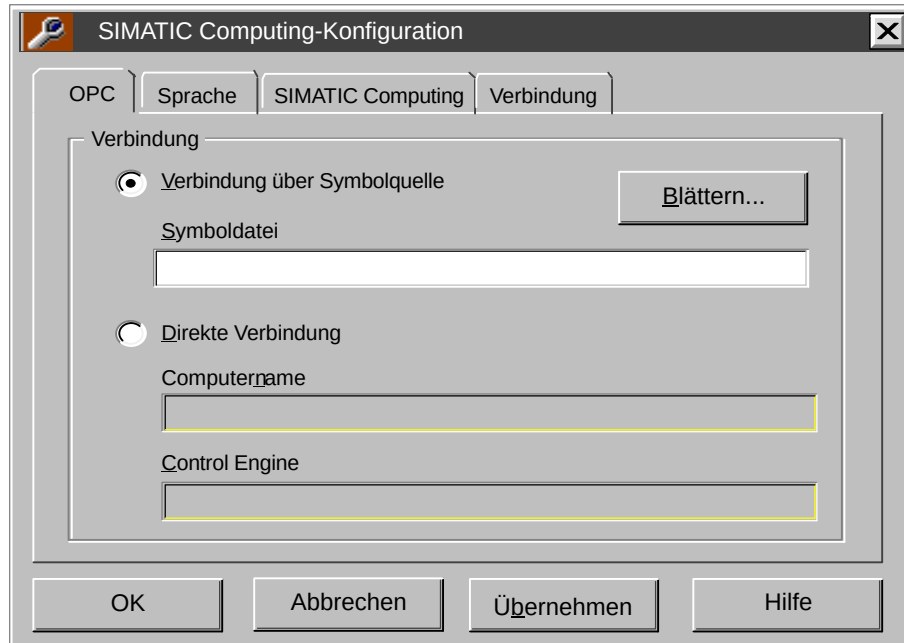


Bild D-1 Einrichten der OPC-Verbindung

OPC-Fehlercodes

Tabelle D-1 führt die Fehlercodes für das OPC-Interface auf. OPC-Methoden geben Fehlercodes an HRESULT (Variable im Hexadezimalformat) aus. In Visual C werden Fehlerbedingungen mit HRESULT bearbeitet. In Visual Basic wird die Fehlerbehebung in das VB-Fehlerobjekt (ERR) geschrieben. Sie müssen in Ihrem VB-Programm Code ergänzen, um auf die Fehlercodes vom OPC-Interface zugreifen zu können.

Tabelle D-1 OPC-Fehlercodes

Fehlercode	Fehler	Beschreibung
0 x 80070057	E_INVALIDARG	Der Wert von einem oder mehreren Parametern ist ungültig. Dieser Fehler wird im allgemeinen für einen spezifischen Fehler eingesetzt, wenn erwartet wird, daß Fehler unwahrscheinlich sind oder leicht zu erkennen sind (z.B. wenn nur ein Parameter vorhanden ist).
0x8007000E	E_OUTOFMEMORY	Es ist nicht genügend Speicher vorhanden, um die angegebene Funktion auszuführen. Dieser Fehler kann auftreten, wenn der Server Speicher zuordnen muß, um die angegebene Funktion zu beenden.
0x0004000D	OPC_E_UNSUPPORTEDRATE	Der Server unterstützt die gewünschte Datenrate nicht. Es wird die am nächsten liegende, verfügbare Datenrate verwendet.
0x0004000E	OPC_E_CLAMP	Ein an WRITE übergebener Wert wurde angenommen, doch ist der Wert begrenzt.
0xC0040001	OPC_E_INVALIDHANDLE	Es wurde ein ungültiger Handle übergeben.
0xC0040002	OPC_E_DUPLICATE	Es wurde ein doppelter Parameter übergeben, doch dies ist nicht zulässig.
0xC0040003	OPC_E_UNKNOWNLCID	Der Server unterstützt die angegebene lokale ID nicht.
0xC0040004	OPC_E_BADTYPE	Der Server kann den übergebenen bzw. benötigten Datentyp nicht in den kanonischen Datentyp für dieses Element umwandeln.
0xC0040005	OPC_E_PUBLIC	Die angeforderte Funktion kann nicht mit einer allgemeinen Gruppe ausgeführt werden.
0xC0040006	OPC_E_BADRIGHTS	Die Zugriffsrechte des Elements lassen die gewünschte Funktion nicht zu.
0xC0040007	OPC_E_UNKNOWNITEMID	Die Definition des Elements ist im Adreßbereich des Servers nicht vorhanden. Dies kann auch bei einem bestehenden Element auftreten, wenn das Element von einer externen Funktion online aus dem Adreßbereich des Servers gelöscht wird.
0xC0040008	OPC_E_INVALIDITEMID	Die Definition des Elements entspricht nicht der Syntax des Servers.
0xC0040009	OPC_E_INVALIDFILTER	Die Zeichenfolge des Filters ist ungültig.
0xC004000A	OPC_E_UNKNOWNPATH	Der Zugriffspfad des Elements ist dem Server nicht bekannt.
0xC004000B	OPC_E_RANGE	Ein an WRITE übergebener Wert ist außerhalb des Bereichs.
0xC004000C	OPC_E_DUPLICATE_NAME	Eine Gruppe mit dem gleichen Namen ist bereits auf dem Server vorhanden.

D.2 Auswählen der Sprache

WinAC bietet Software und Hilfe in drei Sprachen: Deutsch, Englisch und Französisch. Die Menüs und die Hilfe werden in der gewählten Sprache angezeigt. Gehen Sie folgendermaßen vor, um die Spracheinstellung für WinAC zu ändern:

1. Wählen Sie im Windows Startmenü den Befehl **SIMATIC > PC Based Control > SIMATIC Computing-Konfiguration**, um das Dialogfeld "SIMATIC Computing-Konfiguration" aufzurufen.
2. Öffnen Sie im Dialogfeld "SIMATIC Computing-Konfiguration" das Register "Sprache".
3. Wählen Sie die Sprache für das CPU-Panel (Deutsch, Englisch oder Französisch) (siehe Bild D-2).
4. Ändern Sie die Sprache mit der Schaltfläche "Übernehmen".
5. Klicken Sie auf "OK", um das Dialogfeld "Optionen" zu schließen.

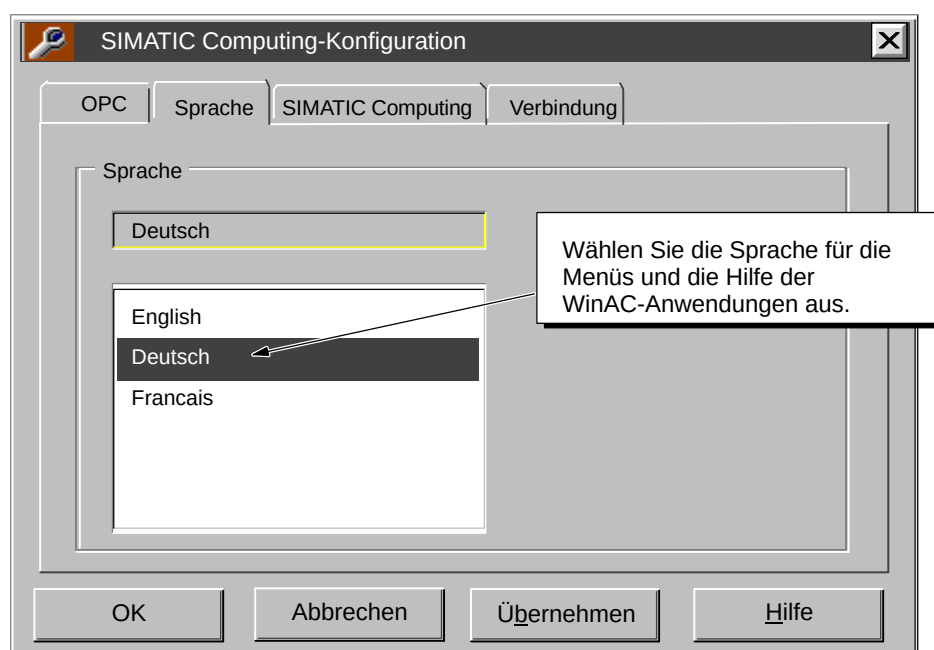


Bild D-2 Einstellen der Sprache für das CPU-Panel und die Hilfe

D.3 Auswählen der Control Engine für ältere Programme

Im Register "SIMATIC Computing" des Konfigurationswerkzeugs können Sie die Kompatibilität zu Programmen herstellen, die mit früheren Versionen von SIMATIC Computing erstellt wurden und über WinAC\Default auf die lokale Control Engine zugegriffen haben. (WinAC\Default kann nicht mit entfernten Computern oder mit mehreren Control Engines eingesetzt werden.) Wenn Sie mit älteren Programmen arbeiten, gehen Sie folgendermaßen vor, um die lokale Control Engine einzustellen:

Hinweis

Dieses Register bietet Kompatibilität mit Anwendungen, die mit älteren Versionen von SIMATIC Computing erstellt wurden und keine Symboldateien unterstützen. Wenn Sie mit Symboldateien arbeiten, müssen Sie dieses Register nicht ausfüllen.

1. Rufen Sie über das Windows Startmenü (**Start > Simatic > PC Based Control > SIMATIC Computing Konfiguration**) das WinAC Konfigurationswerkzeug auf.
2. Öffnen Sie das Register "SIMATIC Computing".
3. Wählen Sie die Control Engine (siehe BildD-3):
 - Aktivieren Sie die Option "WinLC", wenn Sie die WinLC Control Engine einstellen möchten.
 - Aktivieren Sie die Option "CPU416-2 DP ISA", wenn Sie die CPU 416-2 DP ISA Control Engine einstellen möchten.
 - Aktivieren Sie die Option "MPI" und geben Sie eine MPI-Adresse ein, wenn Sie eine SPS im MPI-Netz als Control Engine einstellen möchten.
4. Bestätigen Sie die gewählte Control Engine mit "OK". (Wählen Sie die Schaltfläche "Rückgängig", wenn Sie die Control Engine rücksetzen möchten.)

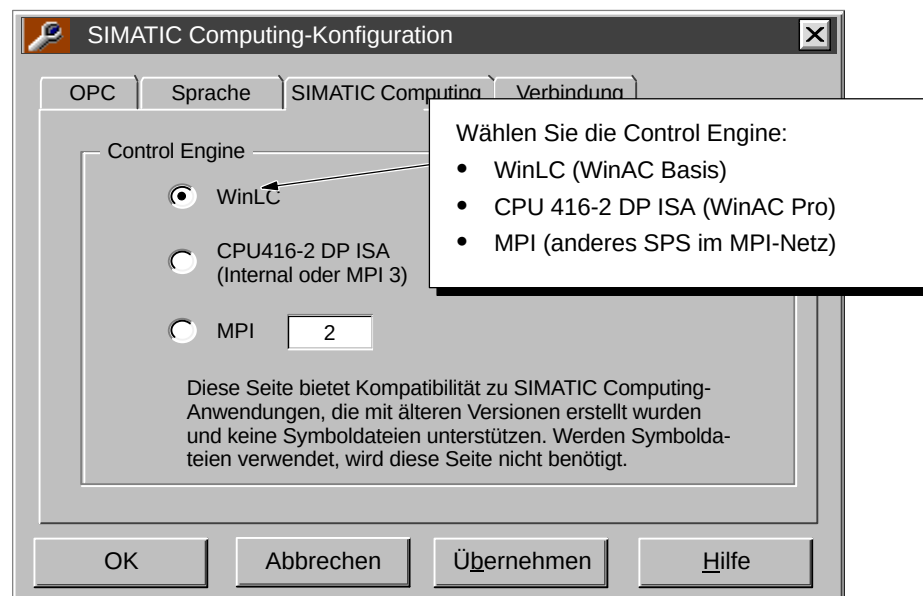


Bild D-3 Einstellen der Control Engine

D.4 Einrichten der Kommunikation in der Anwendung "PG/PC-Schnittstelle einstellen"

In diesem Register können Sie auf die Anwendung "PG/PC-Schnittstelle einstellen" zugreifen, in der Sie die Kommunikation mit WinLC und anderen Automatisierungssystemen im MPI-, PROFIBUS-DP oder H1-Netz einrichten können.

1. Rufen Sie über das Windows Startmenü (**Start > Simatic > PC Based Control > SIMATIC Computing Konfiguration**) das WinAC Konfigurationswerkzeug auf.
2. Öffnen Sie das Register "Connection".

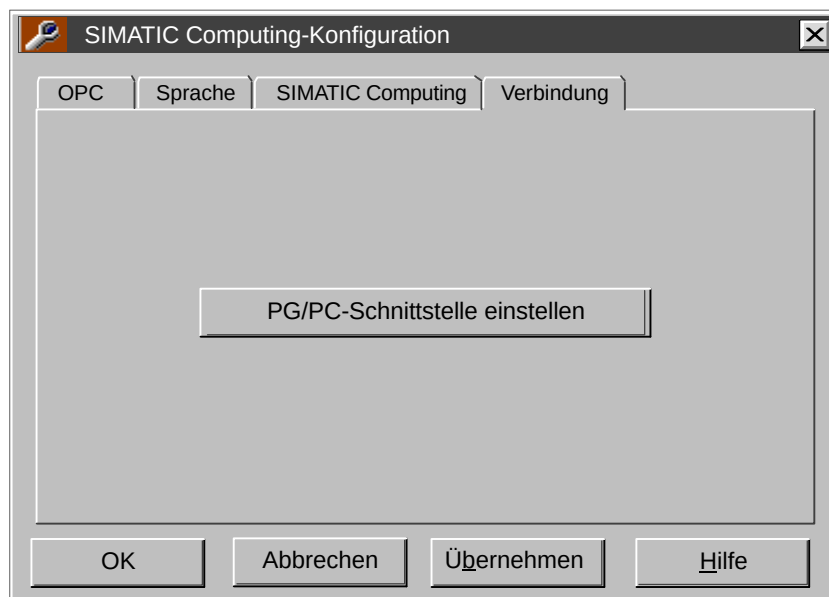


Bild D-4 Zugreifen auf die Anwendung "PG/PC-Schnittstelle einstellen"

Verbinden von STEP 7 mit WinLC auf demselben Computer

1. Führen Sie folgende Arbeitsschritte aus, um STEP 7 als lokalen Zugangspunkt einzurichten:
2. Im aufklappbaren Listefeld "Zugangspunkt der Applikation" wählen Sie **S7ONLINE (STEP 7)** (Bild D-5).
3. Im aufklappbaren Listefeld "Benutzte Schnittstellenparametrierung" wählen Sie **PC Internal (lokal)** als Schnittstellenparameter).

STEP 7 ist jetzt für die Kommunikation mit WinLC auf demselben Computer konfiguriert.

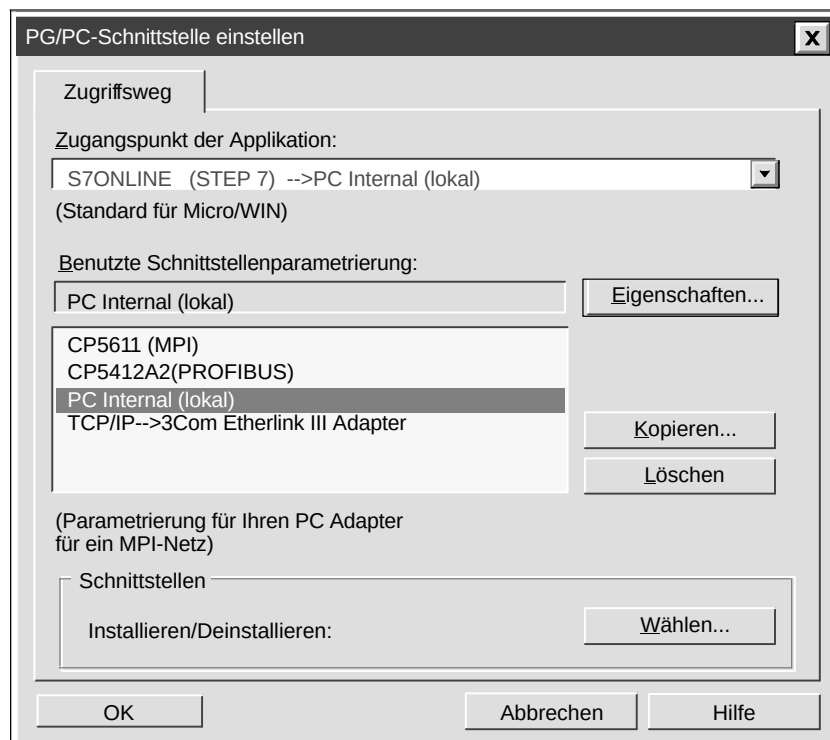


Bild D-5 Einstellen der PG/PC-Schnittstelle für PC Internal (lokal)

Verbinden von STEP 7 mit WinLC auf einem anderen Computer

Zum Verbinden von STEP 7 mit WinLC auf einem anderen Computer müssen Sie die Netzverbindung, über die STEP 7 und WinLC kommunizieren mit der Anwendung "PG/PC-Schnittstelle einstellen" auf dem entfernten Computer einrichten.

Auf dem entfernten Computer muß STEP 7 installiert sein, und auf dem Computer, zu dem Sie die Verbindung herstellen möchten, muß WinLC installiert sein.

Gehen Sie auf dem Computer, auf dem STEP 7 installiert ist, folgendermaßen vor, um STEP 7 für die Kommunikation mit WinLC auf einem entfernten Computer einzurichten:

1. Rufen Sie das Werkzeug zum Einrichten der Schnittstelle im SIMATIC Manager auf. Wählen Sie hierzu den Menübefehl **Extras > PG/PC-Schnittstelle einstellen**.

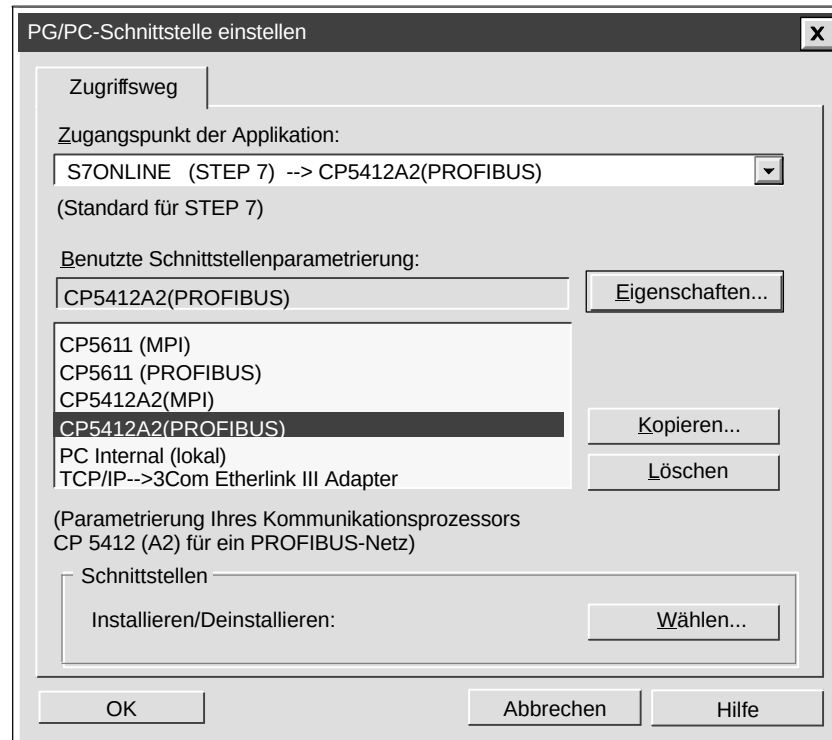


Bild D-6 Einstellen der PG/PC-Schnittstelle auf dem Computer mit STEP 7

2. Im aufklappbaren Listefeld "Zugangspunkt der Applikation" wählen Sie **S7ONLINE (STEP7)**.
3. Wählen Sie die Schnittstellenbeschreibung, die Ihrem Kommunikationspfad entspricht.
 - Bei MPI-Kommunikation wählen Sie eine MPI-Schnittstelle, z.B. **CP5611 (MPI)**.
 - Bei PROFIBUS-DP-Kommunikation wählen Sie eine PROFIBUS-DP-Schnittstelle, z.B. **CP5412A2(PROFIBUS)**.

Die PROFIBUS-Karte von WinLC muß über die Anwendung **PG/PC-Schnittstelle einstellen** komplett eingerichtet werden, bevor WinLC auf anderen PGs im PROFIBUS-DP-Netz sichtbar wird (**S7ONLINE (STEP7) --> Profibus....**). Aktivieren Sie das Kontrollkästchen "PG/PC ist einziger Master am Bus".

- Bei Kommunikation über Industrial Ethernet wählen Sie die TCP/IP-Schnittstelle, z.B. **TCP/IP-> 3Com Etherlink III Ada...** Sie benötigen das Optionspaket NCM für H1-Kommunikation und STEP 7 V5 SP3.

Hinweis

NetPro kann die MPI- und H1-Adressen oder die Busparameter von WinLC nicht auf einem anderen Computer neu einrichten. Die erforderlichen CP-Karten werden nicht von WinLC gesteuert. Diese Einstellungen können Sie nur in der lokalen Anwendung "PG/PC-Schnittstelle einstellen" vornehmen. Die Netzadresse und die Busparameter für PROFIBUS können auf einem anderen Computer erneut konfiguriert werden. WinLC ist der Master der eigenen PROFIBUS-E/A-Karte.

Auf dem Computer, auf dem WinLC installiert ist, müssen Sie den Kommunikationspfad zu Netzen mit Computern, auf denen STEP 7 läuft, einrichten. Zehn Zugangspunkte werden von WinLC installiert. Jeder Zugangspunkt kann auf eine der installierten Schnittstellen weisen.

Beispiel:

```
WinLC_0 --> frei
WinLC_1 --> CP5412A2 (PROFIBUS)
WinLC_2 --> frei
WinLC_3 --> frei
WinLC_4 --> frei
WinLC_5 --> frei
WinLC_6 --> CP5611 (PROFIBUS)
WinLC_7 --> frei
WinLC_8 --> frei
WinLC_9 --> frei
```

In diesem Beispiel kann auf WinLC 3.0 über zwei Karten gleichzeitig zugegriffen werden. WinLC kann über Karten, denen kein Zugangspunkt zugeordnet ist, nicht erreicht werden.

Zum Einrichten einer dieser Zugangspunkte gehen Sie folgendermaßen vor:

1. Rufen Sie das Werkzeug für die Schnittstellenkonfiguration über WinLC auf.
Wählen Sie den Menübefehl (**CPU > PG/PC-Schnittstelle einstellen**).

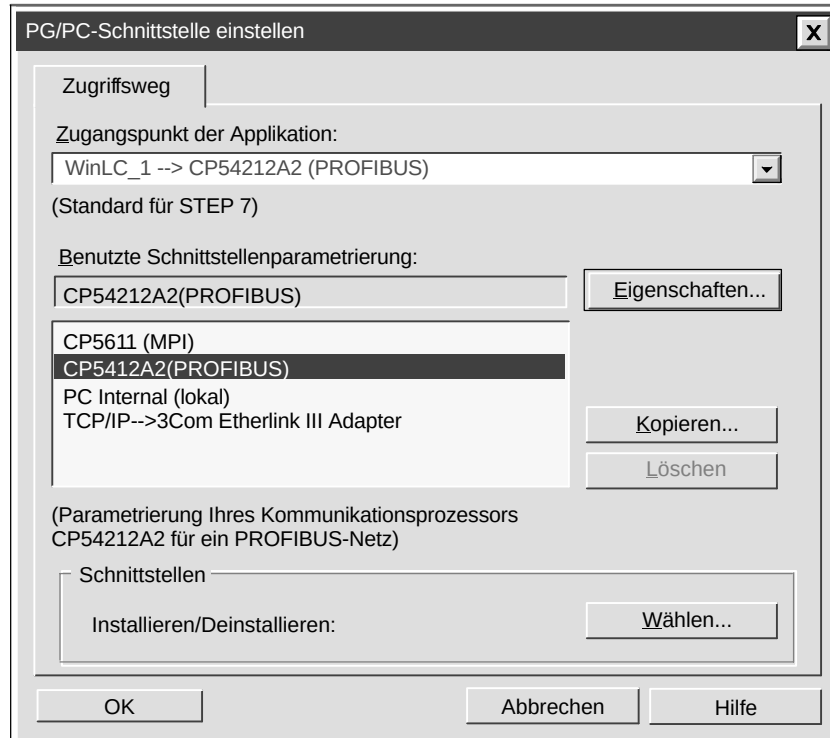


Bild D-7 Einstellen der PG/PC-Schnittstelle in WinLC

2. Im aufklappbaren Listenfeld "Zugangspunkt der Applikation" wählen Sie **WinLC_0**.
3. Wählen Sie die Schnittstellenparametrierung, die Ihrem Kommunikationspfad im Netz entspricht, z.B. **CP5412A2 (PROFIBUS)**.

Wiederholen Sie die Schritte 2 und 3 und konfigurieren Sie alle für die Kommunikation mit einem Netz benötigten Zugangspunkte.

Verbinden von STEP 7 mit Hardware-Automatisierungssystemen

Gehen Sie wie oben unter **Verbinden von STEP 7 mit WinLC auf einem anderen Computer** beschrieben vor. STEP 7 ist jetzt eingerichtet und kann mit WinLC auf einem entfernten Computer sowie mit Hardware-Automatisierungssystemen im Netz kommunizieren. Sie können die gesamte Funktionalität von STEP 7 im Netz nutzen.

Hinweis

PROFIBUS-Busparameter können über WinLC nicht zyklisch verteilt werden.

Einsetzen von SIMATIC Computing mit DCOM



Kapitelübersicht

Mit der Software SIMATIC Computing können Sie über das Distributed Component Object Model (DCOM) von Windows NT im Netz kommunizieren. Über DCOM können Sie dezentrale Anwendungen über ein Netz integrieren. Eine dezentrale Anwendung besteht aus mehreren Prozessen oder unterschiedlichen Computern, die miteinander kooperieren, um eine Aufgabe gemeinsam zu lösen.

DCOM besteht aus Microsoft Konzepten und Programmschnittstellen, mit denen Client-Programme Dienste von Server-Programmen auf anderen Computern in einem Netz anfordern können. Das Component Object Model (COM) bietet einen Satz Schnittstellen, mit denen Clients und Server miteinander auf dem gleichen Computer kommunizieren können (unter Windows 95 und Windows NT).

Hinweis

Die Control Engine muß auf dem Server-Computer installiert sein. Wenn Sie mit den SIMATIC Controls aus SIMATIC Computing auf die Control Engine zugreifen möchten, müssen Sie die Software SIMATIC Computing auf dem Server-Computer und dem Client-Computer installieren.

Kapitel	Beschreibung	Seite
E.1	Entfernter Zugriff über DCOM	E-2
E.2	Einrichten der Berechtigungen für den Server-Computer	E-4
E.3	Einrichten der Berechtigungen für den Client-Computer	E-14
E.4	Fehlerbehebung	E-20

E.1 Entfernter Zugriff über DCOM

DCOM besteht aus Microsoft Konzepten und Programmschnittstellen, mit denen Client-Programme Dienste von Server-Programmen auf anderen Computern in einem Netz anfordern können. Das Component Object Model (COM) bietet einen Satz Schnittstellen, mit denen Clients und Server miteinander auf dem gleichen Computer kommunizieren können (unter Windows 95 und Windows NT).

Wie in Abschnitt 4.1 beschrieben, können Sie SIMATIC Computing auf einem Einzelplatzrechner betreiben (siehe Bild E-1). In diesem Model verfügt der Computer über das komplette Steuerungssystem. Mit SIMATIC Computing können Sie nicht nur auf WinLC von WinAC Basis und auf eine Slot-PLC wie die CPU 416-2 DP ISA von WinAC Pro zugreifen, sondern Sie haben auch Zugriff auf weitere Automatisierungssysteme im Netz.

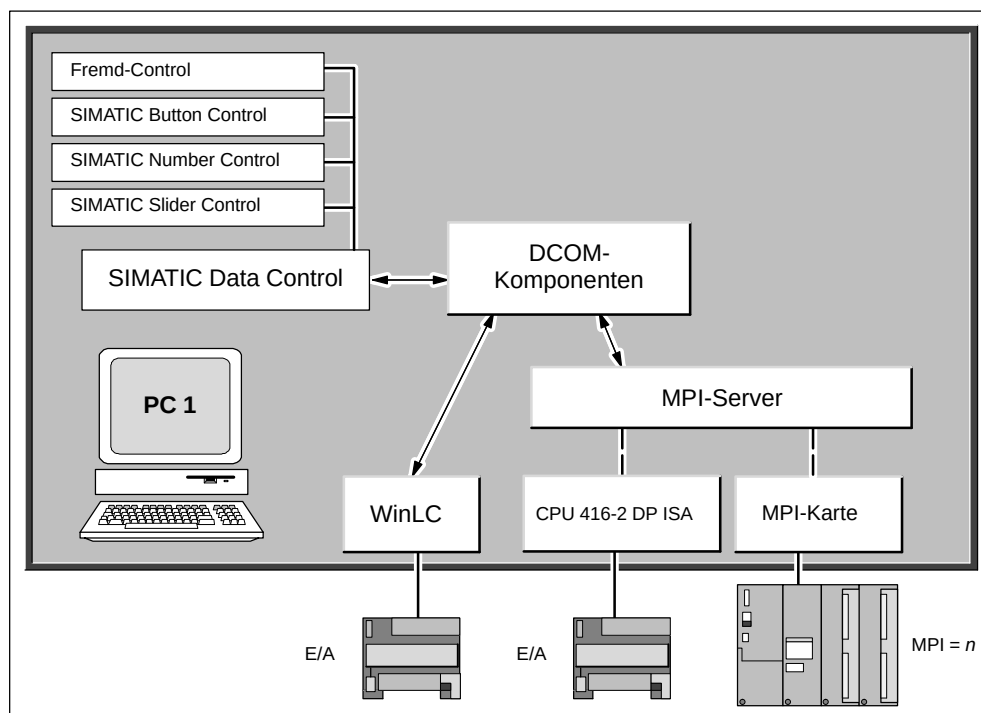


Bild E-1 Einsatz von WinAC auf einem einzelnen Computer

Mit der DCOM-Technologie von Microsoft können Sie ein Netz aus Computern erstellen, die zusammen ein Steuerungssystem für eine Maschine oder einen Prozeß darstellen. Bild E-2 zeigt einen Computer, auf dem eine Anwendung installiert ist, die mit ActiveX Controls (aus SIMATIC Computing) arbeitet und die über DCOM mit einem anderen Computer kommuniziert, auf dem WinLC (oder ein anderes S7-Automatisierungssystem) einen Prozeß steuert.

Das Betriebssystem Windows NT verfügt über ein Konfigurationswerkzeug (dcomcnfg), mit dem Sie Ihr DCOM-Netz einrichten können. Richten Sie mit diesem Werkzeug die Server- und Client-Computer ein. Weitere Informationen zum Einrichten des Server-Computers entnehmen Sie Abschnitt E.2. Weitere Informationen zum Einrichten des Client-Computers entnehmen Sie Abschnitt E.3.

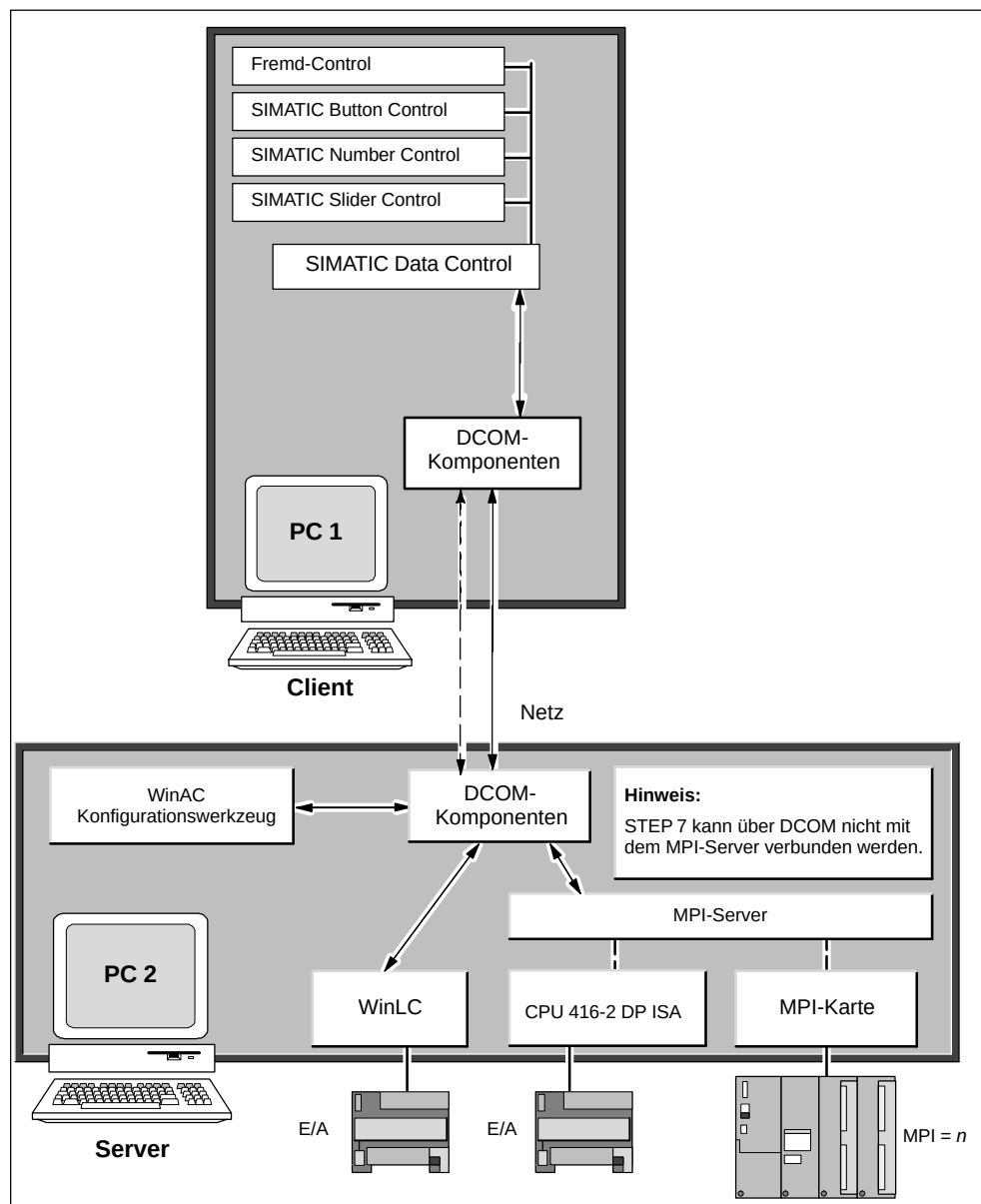


Bild E-2 Arbeiten mit SIMATIC Computing über DCOM

Hinweis

Die Autorisierung für WinAC installieren Sie auf dem Server-Computer, die Autorisierung für SIMATIC Computing installieren Sie auf dem Client-Computer. Wenn Sie SIMATIC Computing auf einem PC betreiben möchten, auf dem WinLC nicht läuft, dann müssen Sie SIMATIC Computing als Standalone-Version erwerben. Ausführliche Informationen zum Installieren der Autorisierung entnehmen Sie dem Abschnitt 3.2.

E.2 Einrichten der Berechtigungen für den Server-Computer

Das DCOM-Netz besteht aus einem Server-Computer (auf dem sich die Control Engine befindet) und einem oder mehreren Client-Computern. Windows NT verfügt über ein Konfigurationswerkzeug zum Einrichten der Netzparameter, z.B. Zugriffsschutz und Zugriffsrechte. Für die Server-Anwendung müssen Sie das Benutzerkonto angeben, das auf die Anwendung zugreifen bzw. sie starten darf, sowie die Benutzerkonten, die dazu dienen, die Anwendung laufen zu lassen. So wird Ihr Prozeß vor unberechtigtem Zugriff geschützt. Bild E-3 führt die wesentlichen Schritte auf, die zum Einrichten des Servers erforderlich sind.



Vorsicht

Ist die Berechtigung für den Zugriff auf die Anwendungen erteilt, können andere Anwender Programme starten und stoppen und auf Dateien in Ihrem Computer zugreifen. Bei nicht eingeschränktem Zugriff für alle Anwender im Netz kann es zu Problemen aufgrund unbeabsichtigter oder beabsichtigter Eingriffe kommen. Schränken Sie den Zugriff immer auf die Anwender ein, die mit den Anwendungen und Dateien auf dem Computer arbeiten müssen.

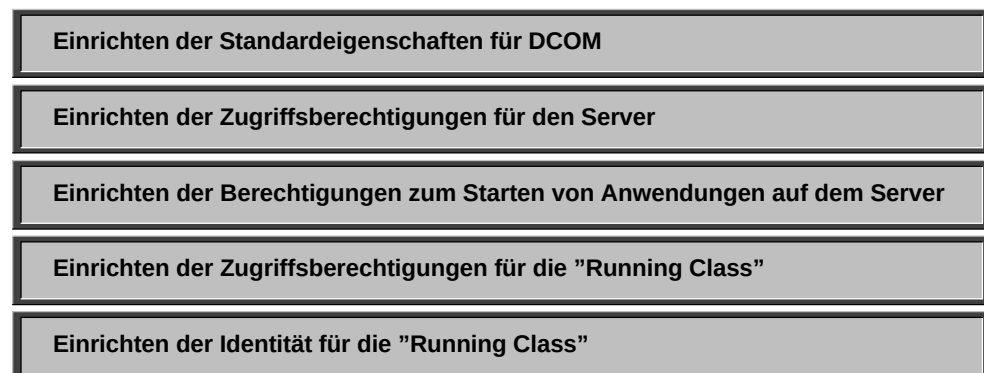


Bild E-3 Schritte zum Konfigurieren des DCOM-Servers

Aufrufen des Editors für die DCOM-Konfiguration

Zum Einrichten des DCOM-Servers müssen Sie das Werkzeug für die DCOM-Konfiguration auf dem Computer, der als Server dienen soll, starten. Gehen Sie folgendermaßen vor, um das Werkzeug für die DCOM-Konfiguration aufzurufen:

1. Wählen Sie den Menübefehl **Start > Ausführen...** im Windows Startmenü.
2. Geben Sie im Dialogfeld "Ausführen" **dcomcnfg** ein und bestätigen Sie mit "OK".

Daraufhin zeigt das Werkzeug für die DCOM-Konfiguration das Dialogfeld "Eigenschaften von DCOM-Konfiguration" an.

Einrichten der Standardeigenschaften für die DCOM-Kommunikation

Die Eigenschaften des Computers für DCOM richten Sie im Dialogfeld "Eigenschaften von DCOM-Konfiguration" ein (siehe Bild E-4).

1. Öffnen Sie das Register "Standardeigenschaften".
2. Wählen Sie die Option "DCOM (Distributed COM) auf diesem Computer aktivieren".
3. Stellen Sie für den "Standard-Authentication-Level" die Option "Verbinden" ein.
4. Stellen Sie für den "Standard-Impersonation-Level" die Option "Identifizieren" ein.

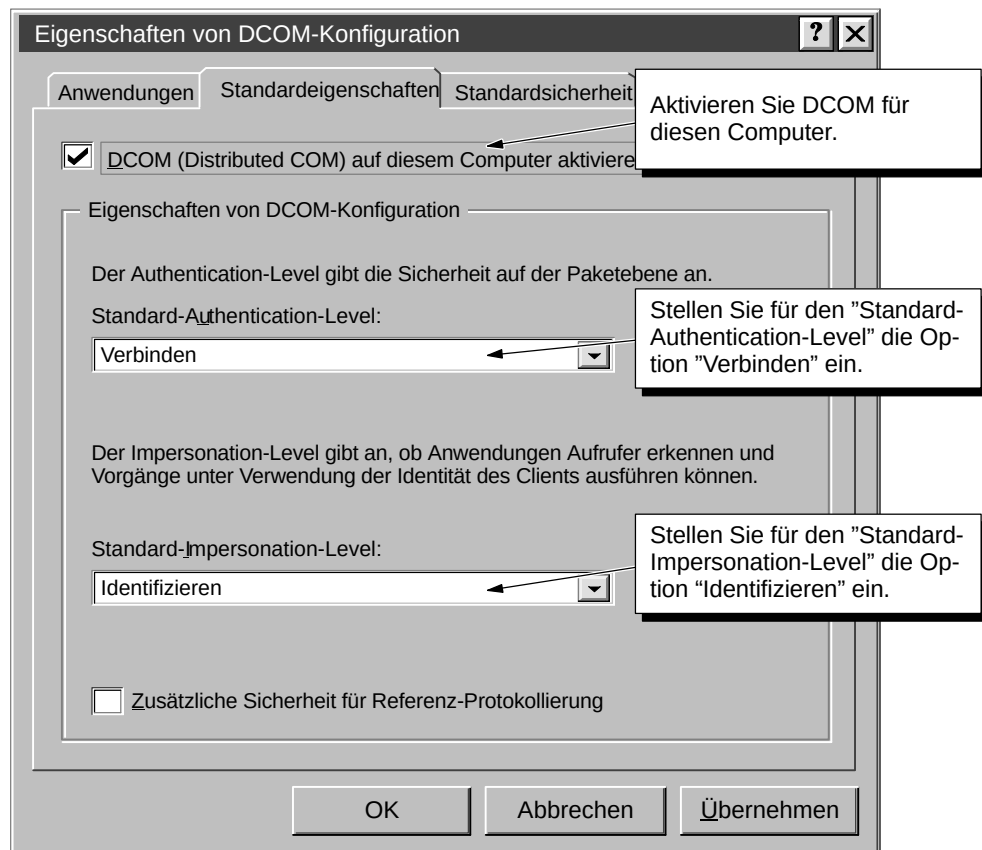


Bild E-4 Eigenschaften der DCOM-Konfiguration

Einrichten der Berechtigungen für den Zugriff auf Software auf dem Server

1. Öffnen Sie das Register "Standardsicherheit", um die Optionen für den Zugriffsschutz in DCOM anzuzeigen (siehe Bild E-5).
2. Klicken Sie auf die Schaltfläche "Standard ändern" für "Standard-Zugriffsberechtigungen", um das Dialogfeld "Registrierungswertberechtigungen" anzuzeigen.

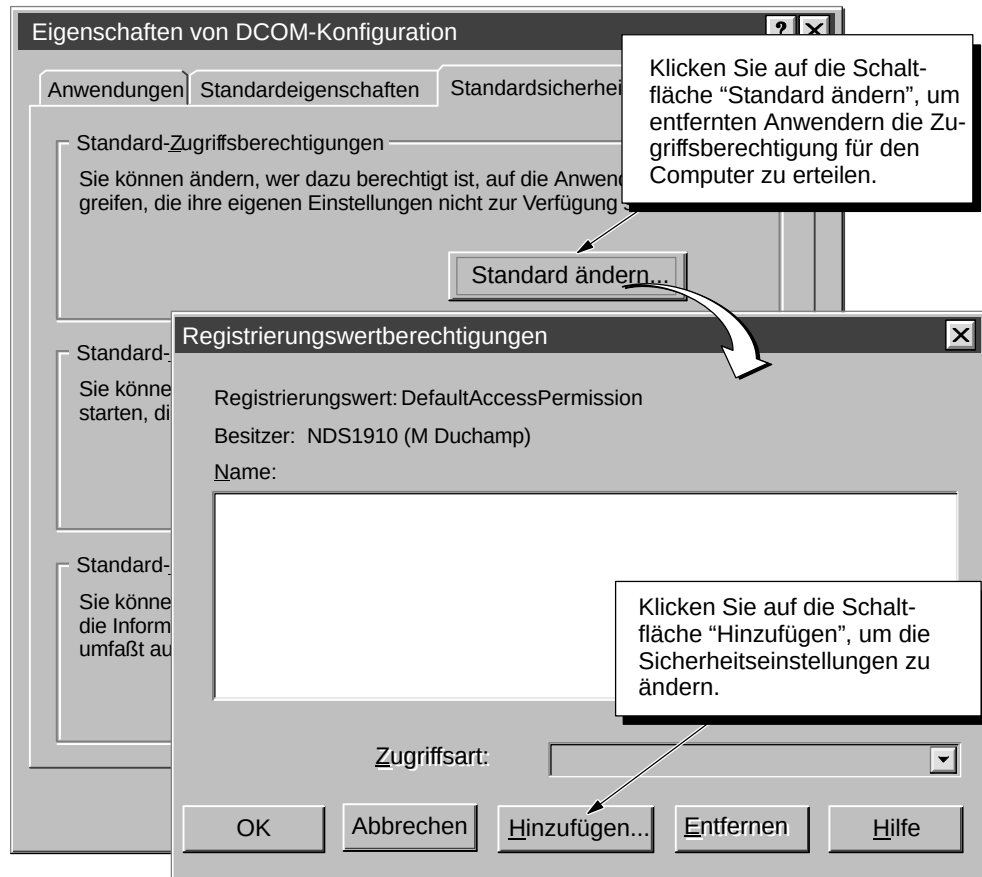


Bild E-5 Einrichten der Standardzugriffsberechtigungen für DCOM

3. Wählen Sie die Schaltfläche "Hinzufügen", um das Dialogfeld "Benutzer und Gruppen hinzufügen" aufzurufen und die Sicherheitseinstellungen für den Zugriff auf den Server zu ändern (siehe Bild E-6).
4. Wählen Sie im Feld "Namen" die Option "Everyone" (oder die entsprechende Benutzergruppe) und klicken Sie auf die Schaltfläche "Hinzufügen".
5. Wählen Sie "INTERACTIVE" und klicken Sie auf die Schaltfläche "Hinzufügen".
6. Wählen Sie "SYSTEM" und klicken Sie auf die Schaltfläche "Hinzufügen".

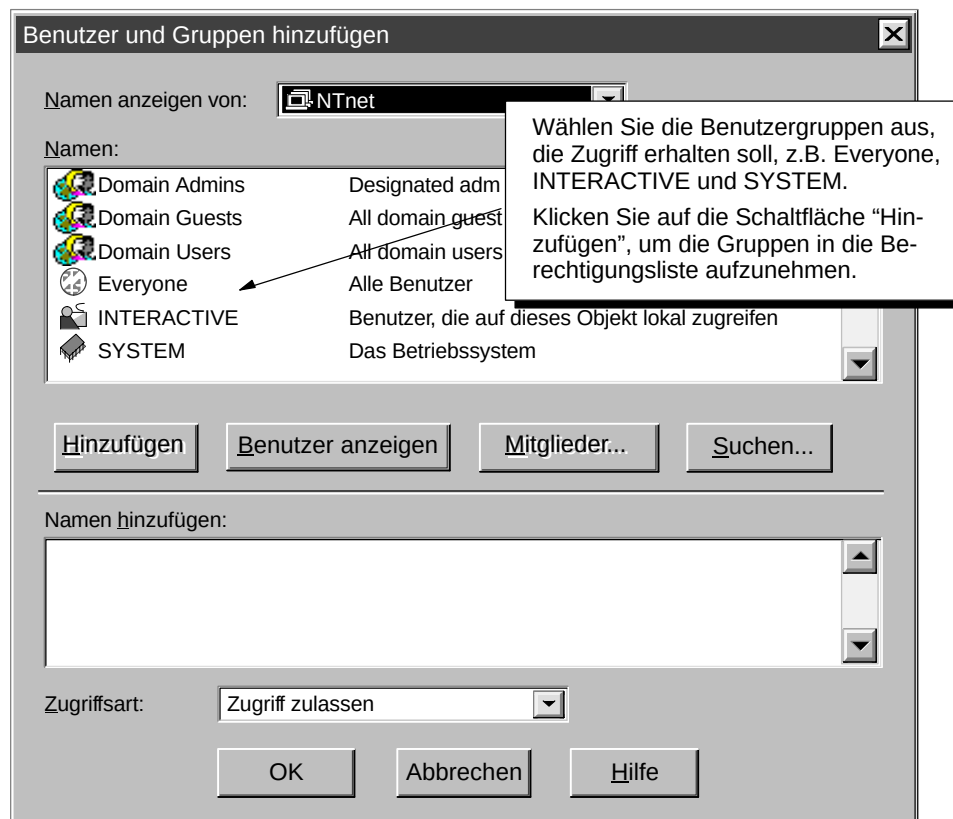


Bild E-6 Ändern der Zugriffsberechtigungen für Benutzer und Gruppen

7. Bestätigen Sie Ihre Eingaben mit "OK", die Änderungen werden daraufhin in das Dialogfeld "Registrierungswertberechtigungen" übernommen.
8. Wählen Sie im Dialogfeld "Registrierungswertberechtigungen" die Schaltfläche "OK", um die Änderungen an den Standardzugriffsberechtigungen einzugeben. Das Dialogfeld "Registrierungswertberechtigungen" wird geschlossen und zeigt wieder das Dialogfeld "Eigenschaften von DCOM-Konfiguration" an (Bild E-5).

**Vorsicht**

Ist die Berechtigung für den Zugriff auf die Anwendungen erteilt, können andere Anwender (z.B. "Everyone") Programme starten und stoppen und auf Dateien in Ihrem Computer zugreifen. Bei nicht eingeschränktem Zugriff für alle Anwender im Netz kann es zu Problemen aufgrund unbeabsichtigter oder beabsichtigter Eingriffe kommen. Schränken Sie den Zugriff immer auf die Anwender ein, die mit den Anwendungen und Dateien auf dem Computer arbeiten müssen.

Einrichten der Berechtigungen zum Starten von Software auf dem Server

1. Klicken Sie auf die Schaltfläche "Standard ändern" für "Standard-Startberechtigungen", um das Dialogfeld "Registrierungswertberechtigungen" anzuzeigen (siehe Bild E-7).
2. Wählen Sie die Schaltfläche "Hinzufügen", um das Dialogfeld "Benutzer und Gruppen hinzufügen" aufzurufen und die Sicherheitseinstellungen für den Zugriff auf den Server zu ändern (siehe Bild E-8).

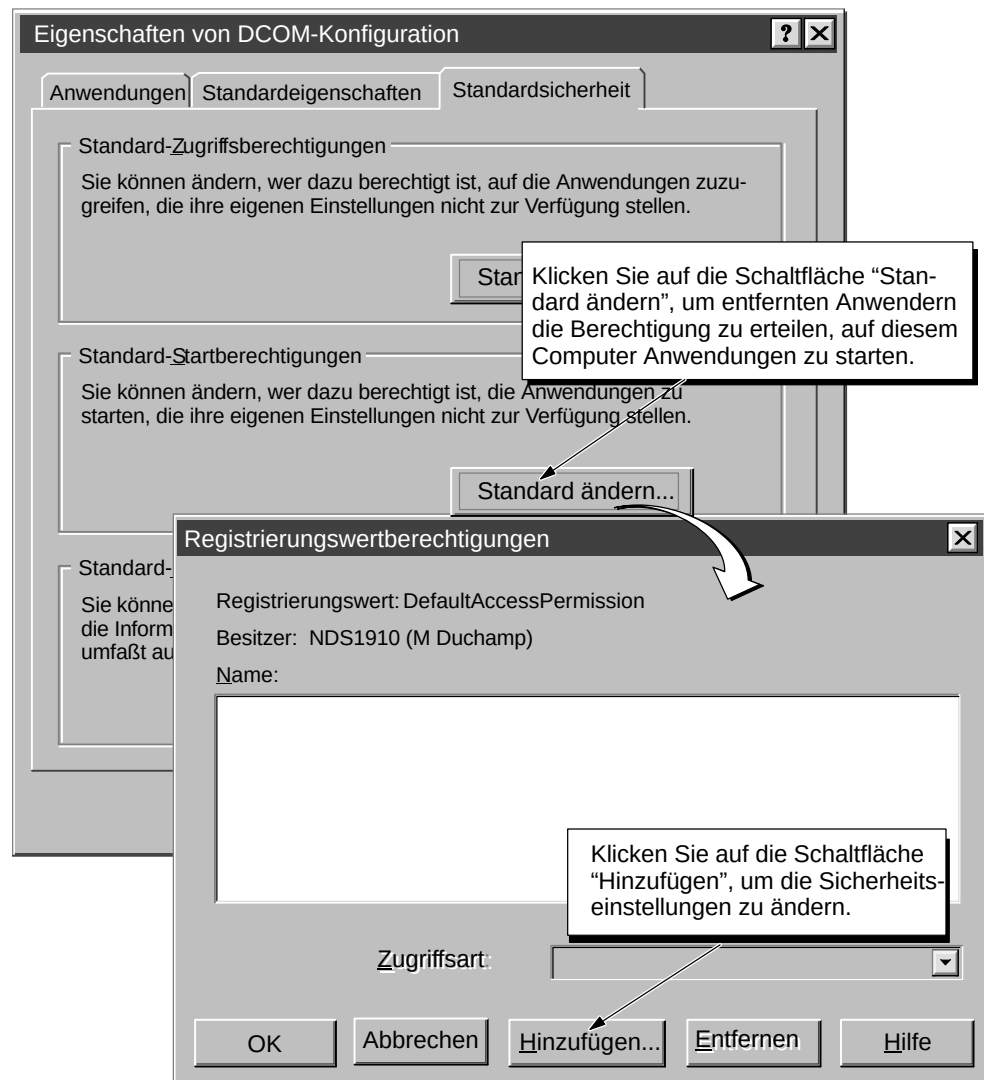


Bild E-7 Einrichten der Standard-Startberechtigungen für DCOM

3. Wählen Sie im Feld "Namen" im Dialogfeld "Benutzer und Gruppen hinzufügen" (Bild E-8) die Option "Everyone" (oder die gewünschte Benutzergruppe) und klicken Sie auf die Schaltfläche "Hinzufügen".



Vorsicht

Ist die Berechtigung für den Zugriff auf die Anwendungen erteilt, können andere Anwender (z.B. "Everyone") Programme starten und stoppen und auf Dateien in Ihrem Computer zugreifen. Bei nicht eingeschränktem Zugriff für alle Anwender im Netz kann es zu Problemen aufgrund unbeabsichtigter oder beabsichtigter Eingriffe kommen. Schränken Sie den Zugriff immer auf die Anwender ein, die mit den Anwendungen und Dateien auf dem Computer arbeiten müssen.

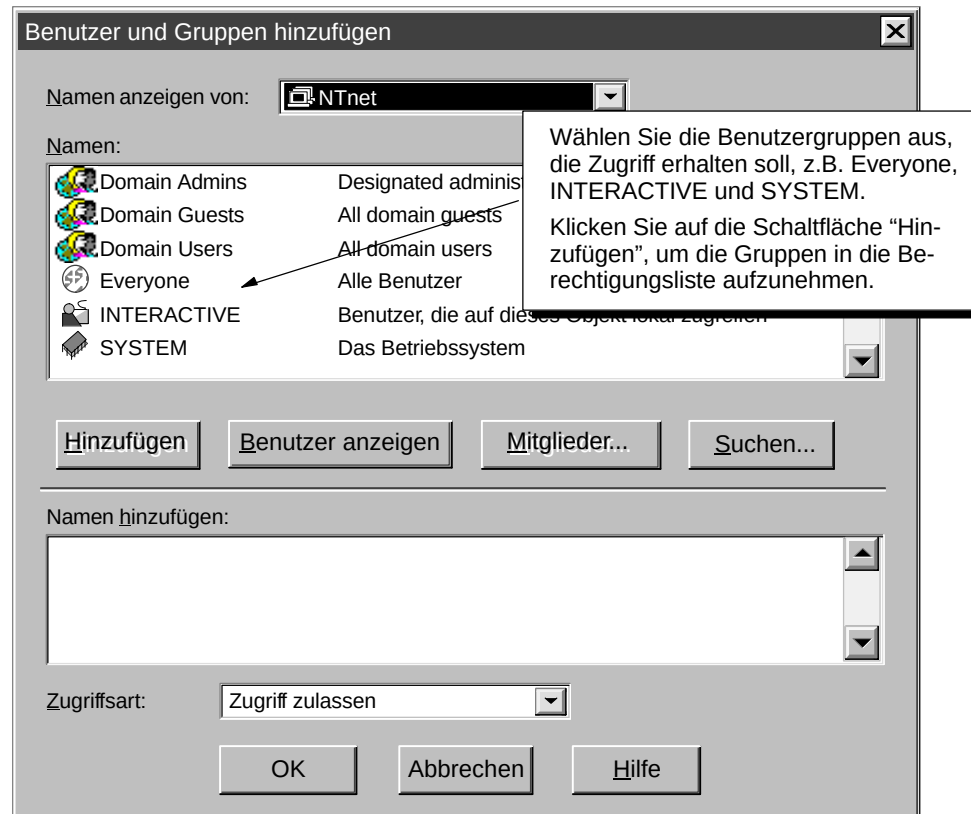


Bild E-8 Ändern der Startberechtigungen für Benutzer und Gruppen

4. Wählen Sie "INTERACTIVE" und klicken Sie auf die Schaltfläche "Hinzufügen".
5. Wählen Sie "SYSTEM" und klicken Sie auf die Schaltfläche "Hinzufügen".
6. Bestätigen Sie Ihre Eingaben mit "OK", die Änderungen werden daraufhin in das Dialogfeld "Registrierungswertberechtigungen" übernommen.
7. Wählen Sie im Dialogfeld "Registrierungswertberechtigungen" die Schaltfläche "OK", um die Änderungen an den Standardzugriffsberechtigungen einzugeben. Das Dialogfeld "Registrierungswertberechtigungen" wird geschlossen und zeigt wieder das Dialogfeld "Eigenschaften von DCOM-Konfiguration" an.

Einrichten der Eigenschaften für "Running Class"

Gehen Sie folgendermaßen vor, um die Eigenschaften für die "Running Class" des Servers einzurichten:

1. Öffnen Sie das Register "Anwendungen" im Dialogfeld "Eigenschaften von DCOM-Konfiguration (siehe Bild E-9).
2. Wählen Sie "Running Class" in der Liste der Anwendungen aus.
3. Wählen Sie die Schaltfläche "Eigenschaften", um das Dialogfeld "Eigenschaften von Running Class" aufzurufen.

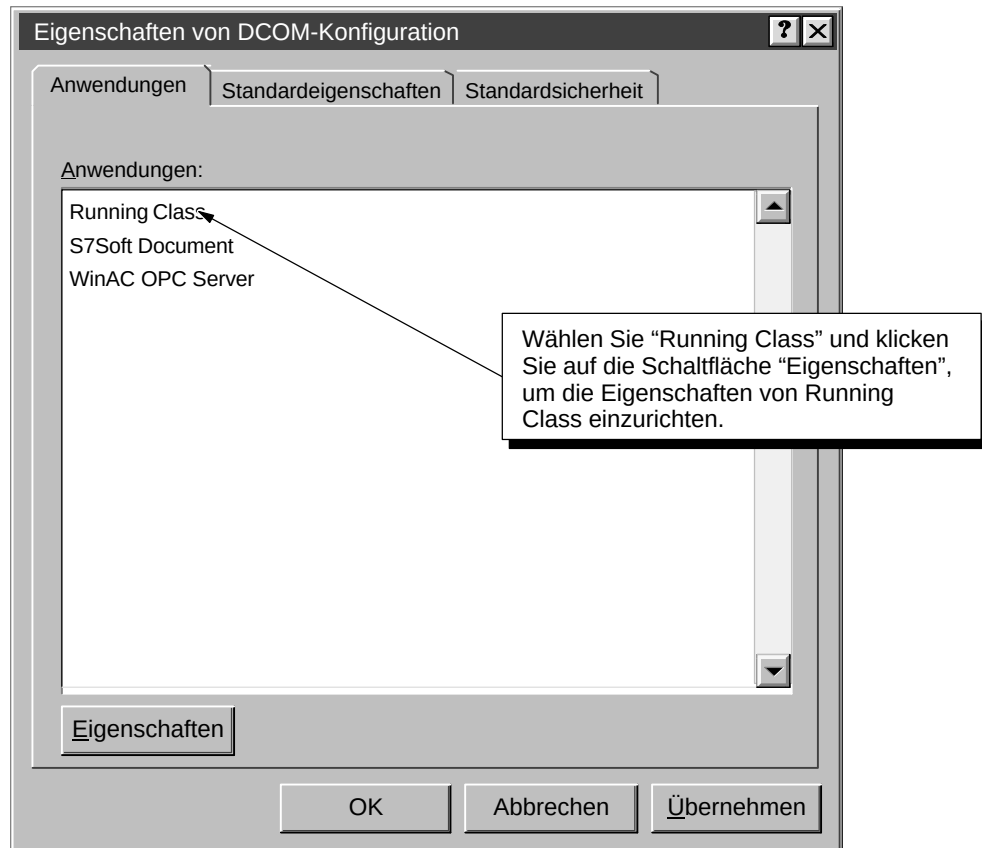


Bild E-9 Einstellen der Running Class für DCOM

Einrichten der Zugriffsberechtigungen für die "Running Class"



Vorsicht

Ist die Berechtigung für den Zugriff auf die Anwendungen erteilt, können andere Anwender (z.B. "Everyone") Programme starten und stoppen und auf Dateien in Ihrem Computer zugreifen. Bei nicht eingeschränktem Zugriff für alle Anwender im Netz kann es zu Problemen aufgrund unbeabsichtigter oder beabsichtigter Eingriffe kommen. Schränken Sie den Zugriff immer auf die Anwender ein, die mit den Anwendungen und Dateien auf dem Computer arbeiten müssen.

Gehen Sie folgendermaßen vor, um die Zugriffsberechtigungen für die "Running Class" einzurichten:

1. Öffnen Sie das Register "Sicherheit" im Dialogfeld "Eigenschaften von Running Class".
2. Wählen Sie "Standard-Zugriffsberechtigungen verwenden" und klicken Sie auf die Schaltfläche "Ändern" (siehe Bild E-10).
3. Ist die Option "Everyone" (oder die gewünschte Benutzergruppe) nicht in der Liste "Namen" im Dialogfeld "Berechtigungen" enthalten, klicken Sie auf die Schaltfläche "Hinzufügen", um das Dialogfeld "Benutzer und Gruppen hinzufügen" aufzurufen (siehe Bild E-11).

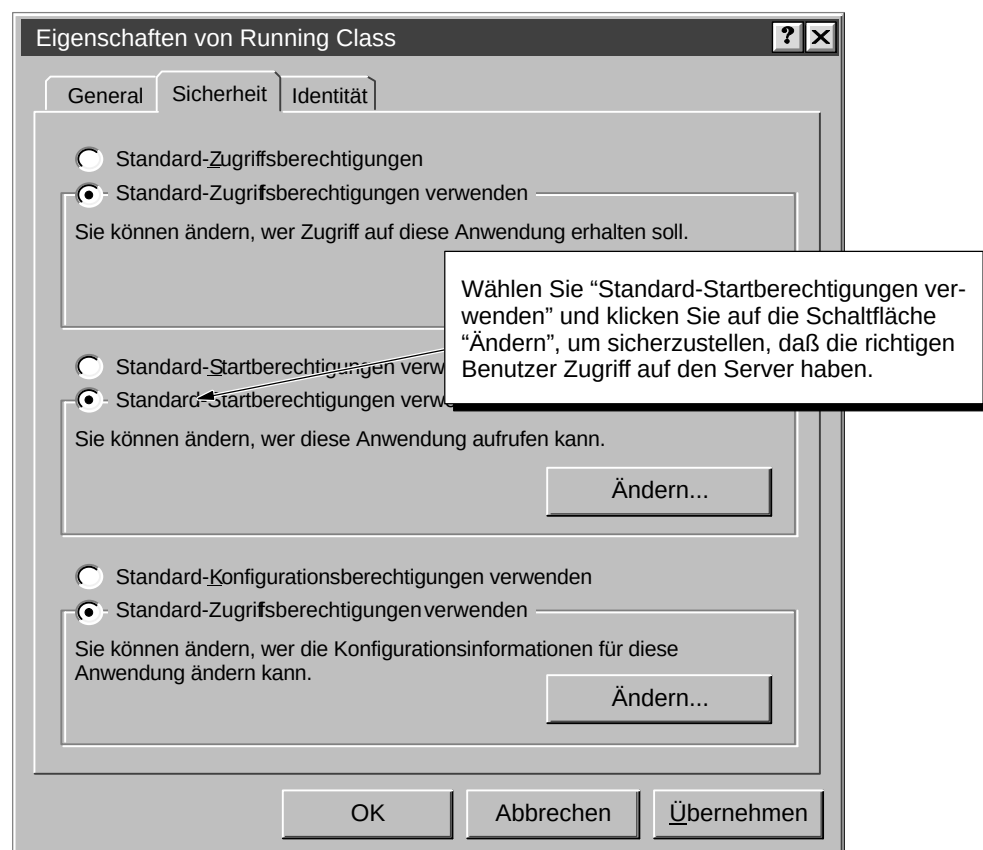


Bild E-10 Einrichten der DCOM-Zugriffsberechtigungen für den Server

4. Ergänzen Sie im Dialogfeld "Benutzer und Gruppen hinzufügen" (Bild E-11) Benutzer/Gruppen wie gewünscht.
5. Klicken Sie auf die Schaltfläche "OK", um zum Dialogfeld "Eigenschaften von Running Class" zurückzukehren.

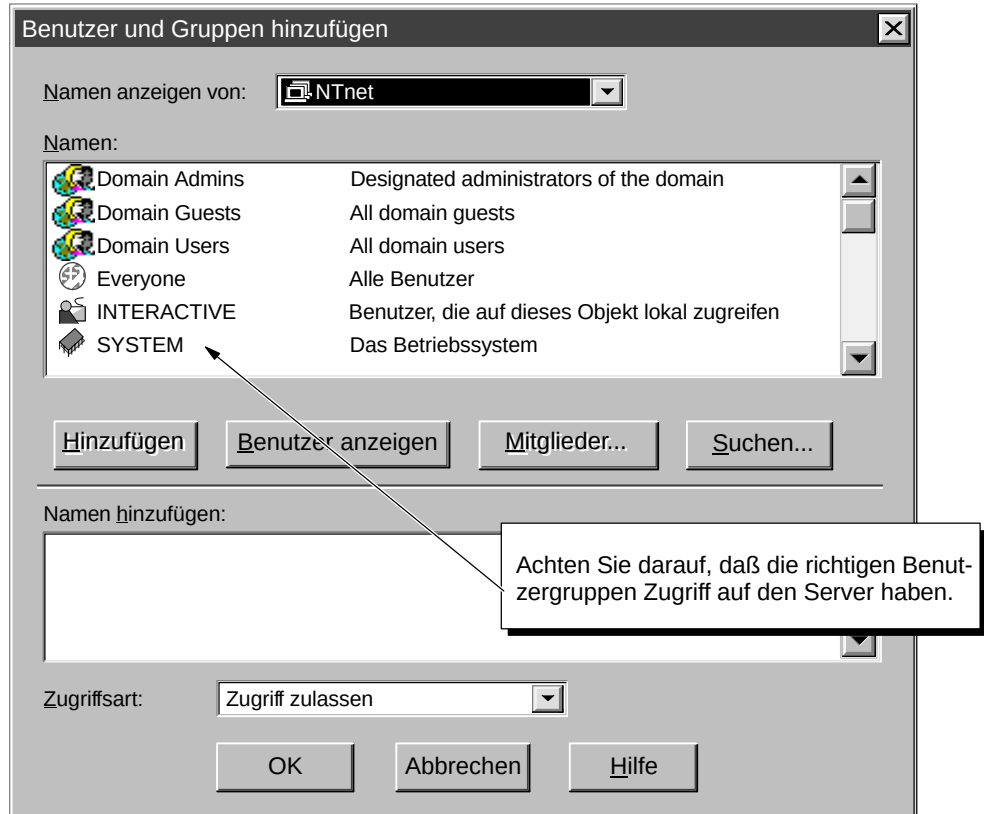


Bild E-11 Ändern der Zugriffsberechtigungen für Benutzer und Gruppen

Einrichten der Identität für die "Running Class"

Gehen Sie folgendermaßen vor, um die Identitätsberechtigungen für die "Running Class" einzurichten:

- Öffnen Sie das Register "Identität" und wählen Sie den Benutzer aus, der Zugriff auf den Server haben soll (siehe Bild E-12).
 - Läuft die Control Engine (z.B. WinLC) als NT-Dienst, wählen Sie "Dieser Benutzer" und geben den DOMAIN/LOGIN-Namen des Benutzers ein, in dessen Sicherheitsumgebung die Anwendung laufen soll (das ist nicht der entfernte Benutzer, sondern das Benutzerkonto, auf dem die Server-Anwendung läuft). Geben Sie das Paßwort für Domain und Benutzer ein.
 - Läuft die Control Engine nicht als NT-Dienst, wählen Sie "Der interaktive Benutzer" (das ist der Benutzer, der aktuell auf dem Computer eingeloggt ist).
- Bestätigen Sie die Identitätsberechtigungen für die Running Class mit "OK".

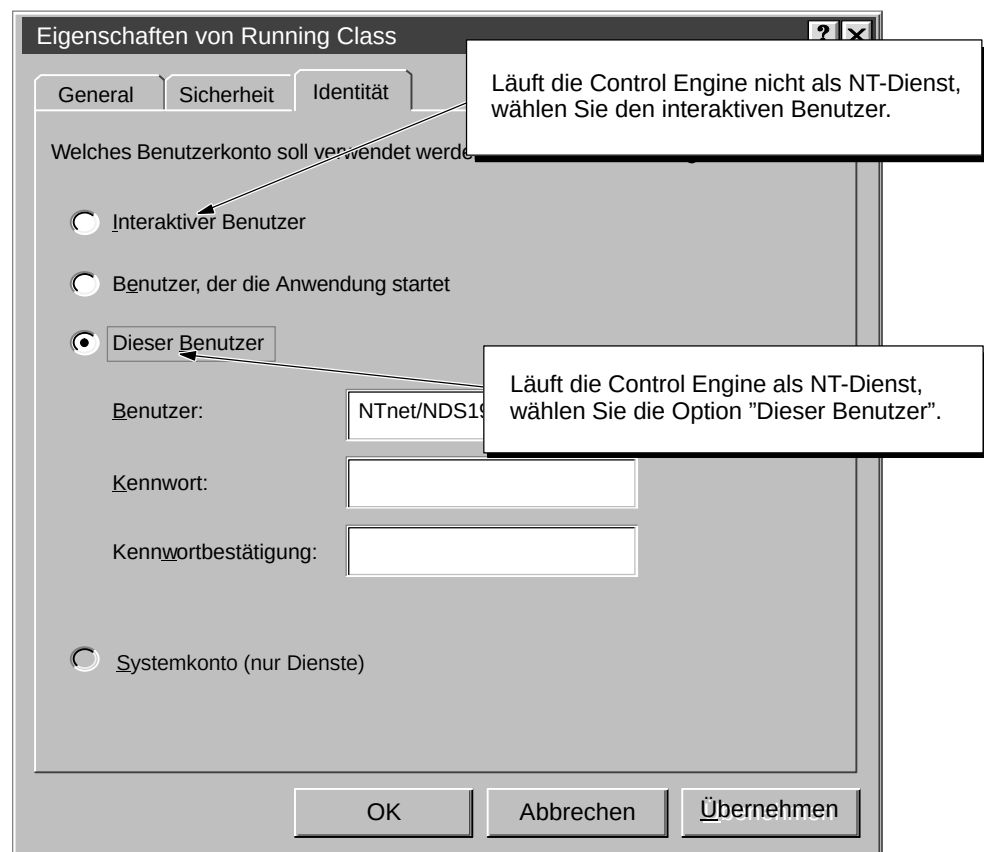


Bild E-12 Einrichten der DCOM-Identitätsberechtigungen für den Server

E.3 Einrichten der Berechtigungen für den Client-Computer

Bevor Sie SIMATIC Computing mit DCOM einsetzen können, müssen Sie anhand der DCOM-Konfiguration die Eigenschaften der Anwendung, z.B. Zugriffsschutz und Ablage, einstellen. Auf einem Computer, auf dem die Client-Anwendung läuft (die Anwendung, die eine Anforderung an eine Server-Anwendung initiiert), müssen Sie die Ablage der Server-Anwendung (die Anwendung, die auf die Anforderung der Clients reagiert), auf die zugegriffen bzw. die gestartet wird, angeben. Bild E-13 führt die wesentlichen Schritte auf, die zum Einrichten des Servers erforderlich sind.

Hinweis

Die Eigenschaften für die "Running Class" richten Sie nicht auf dem Client-Computer ein. Die Running Class definieren Sie auf dem Server-Computer (siehe Bild E-9).

Einrichten der Standardeigenschaften für DCOM

Einrichten der Berechtigungen für den Zugriff auf Software auf dem Client

Einrichten der Berechtigungen zum Starten von Anwendungen auf dem Client

Bild E-13 Schritte zum Konfigurieren des DCOM-Clients



Vorsicht

Ist die Berechtigung für den Zugriff auf die Anwendungen erteilt, können andere Anwender (z.B. "Everyone") Programme starten und stoppen und auf Dateien in Ihrem Computer zugreifen. Bei nicht eingeschränktem Zugriff für alle Anwender im Netz kann es zu Problemen aufgrund unbeabsichtigter oder beabsichtigter Eingriffe kommen. Schränken Sie den Zugriff immer auf die Anwender ein, die mit den Anwendungen und Dateien auf dem Computer arbeiten müssen.

Aufrufen des Editors für die DCOM-Konfiguration

Zum Einrichten des DCOM-Client müssen Sie das Werkzeug für die DCOM-Konfiguration auf dem Computer, der als Client dienen soll, starten. Gehen Sie folgendermaßen vor, um das Werkzeug für die DCOM-Konfiguration aufzurufen:

1. Wählen Sie den Menübefehl **Start > Ausführen...** im Windows Startmenü.
2. Geben Sie im Dialogfeld "Ausführen" **dcomcnfg** ein und bestätigen Sie mit "OK".

Daraufhin zeigt das Werkzeug für die DCOM-Konfiguration das Dialogfeld "Eigenschaften von DCOM-Konfiguration" an.

Einrichten der Standardeigenschaften für die DCOM-Kommunikation

Die Eigenschaften des Computers für DCOM richten Sie im Dialogfeld "Eigenschaften von DCOM-Konfiguration" ein.

1. Öffnen Sie das Register "Standardeigenschaften". (siehe Bild E-14).
2. Wählen Sie die Option "DCOM (Distributed COM) auf diesem Computer aktivieren".
3. Stellen Sie für den "Standard-Authentication-Level" die Option "Verbinden" ein.
4. Stellen Sie für den "Standard-Impersonation-Level" die Option "Identifizieren" ein.

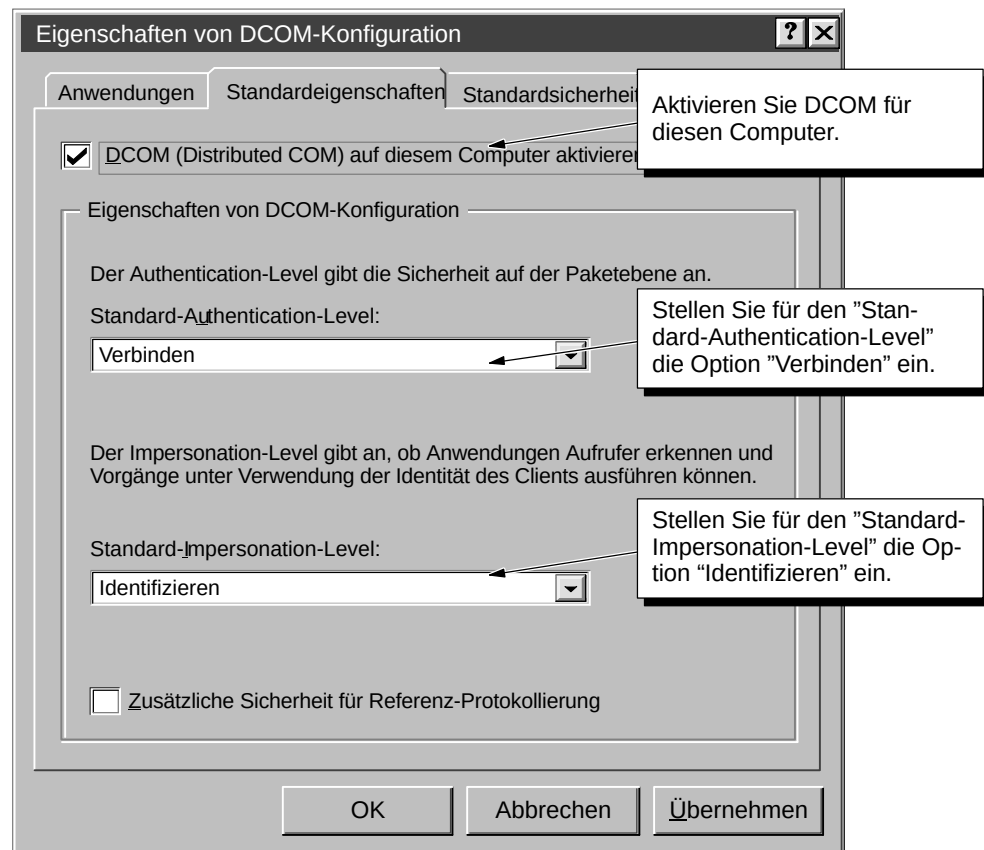


Bild E-14 Eigenschaften der DCOM-Konfiguration

Einrichten der Berechtigungen für den Zugriff auf Software auf dem Client

1. Öffnen Sie das Register "Standardsicherheit", um die Optionen für den Zugriffsschutz in DCOM anzuzeigen (siehe Bild E-15).
2. Klicken Sie auf die Schaltfläche "Standard ändern" für "Standard-Zugriffsberechtigungen", um das Dialogfeld "Registrierungswertberechtigungen" anzuzeigen.

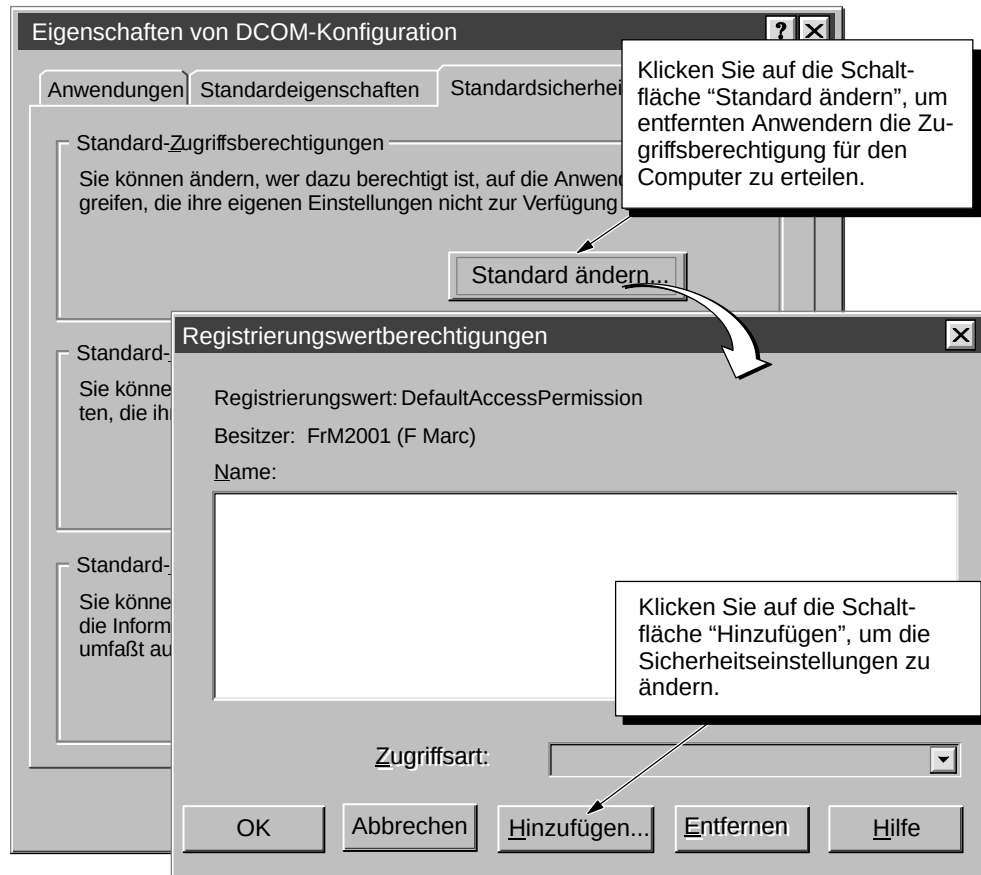


Bild E-15 Einrichten der Standardzugriffsberechtigungen für DCOM

3. Wählen Sie die Schaltfläche "Hinzufügen", um das Dialogfeld "Benutzer und Gruppen hinzufügen" aufzurufen und die Sicherheitseinstellungen für den Zugriff auf den Server zu ändern (siehe Bild E-16).
4. Wählen Sie im Feld "Namen" die Option "Everyone" (oder die entsprechende Benutzergruppe) und klicken Sie auf die Schaltfläche "Hinzufügen".
5. Wählen Sie "INTERACTIVE" und klicken Sie auf die Schaltfläche "Hinzufügen".
6. Wählen Sie "SYSTEM" und klicken Sie auf die Schaltfläche "Hinzufügen".

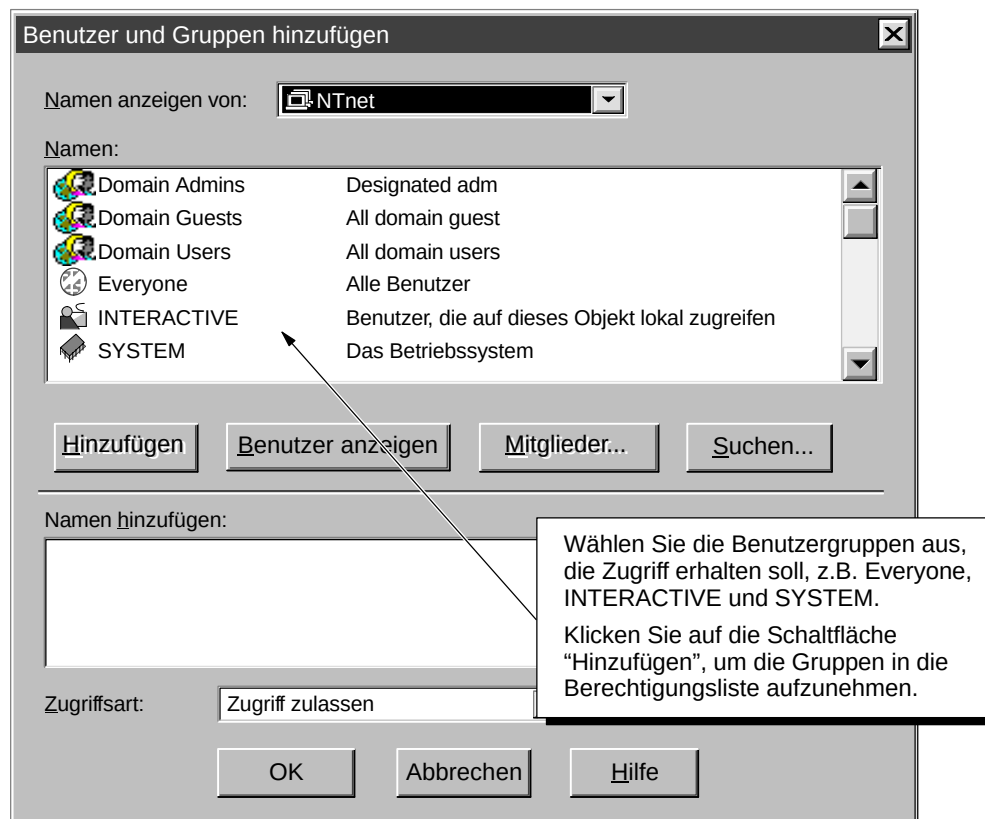


Bild E-16 Ändern der Zugriffsberechtigungen für Benutzer und Gruppen

7. Bestätigen Sie Ihre Eingaben mit "OK", die Änderungen werden daraufhin in das Dialogfeld "Registrierungswertberechtigungen" übernommen.
8. Wählen Sie im Dialogfeld "Registrierungswertberechtigungen" die Schaltfläche "OK", um die Änderungen an den Standardzugriffsberechtigungen einzugeben. Das Dialogfeld "Registrierungswertberechtigungen" wird geschlossen und zeigt wieder das Dialogfeld "Eigenschaften von DCOM-Konfiguration" an (Bild E-15).

**Vorsicht**

Ist die Berechtigung für den Zugriff auf die Anwendungen erteilt, können andere Anwender (z.B. "Everyone") Programme starten und stoppen und auf Dateien in Ihrem Computer zugreifen. Bei nicht eingeschränktem Zugriff für alle Anwender im Netz kann es zu Problemen aufgrund unbeabsichtigter oder beabsichtigter Eingriffe kommen. Schränken Sie den Zugriff immer auf die Anwender ein, die mit den Anwendungen und Dateien auf dem Computer arbeiten müssen.

Einrichten der Berechtigungen zum Starten von Software auf dem Client

1. Klicken Sie auf die Schaltfläche "Standard ändern" für "Standard-Startberechtigungen", um das Dialogfeld "Registrierungswertberechtigungen" anzuzeigen (siehe Bild E-17).
2. Wählen Sie die Schaltfläche "Hinzufügen", um das Dialogfeld "Benutzer und Gruppen hinzufügen" aufzurufen und die Sicherheitseinstellungen für den Zugriff auf den Server zu ändern (siehe Bild E-18).

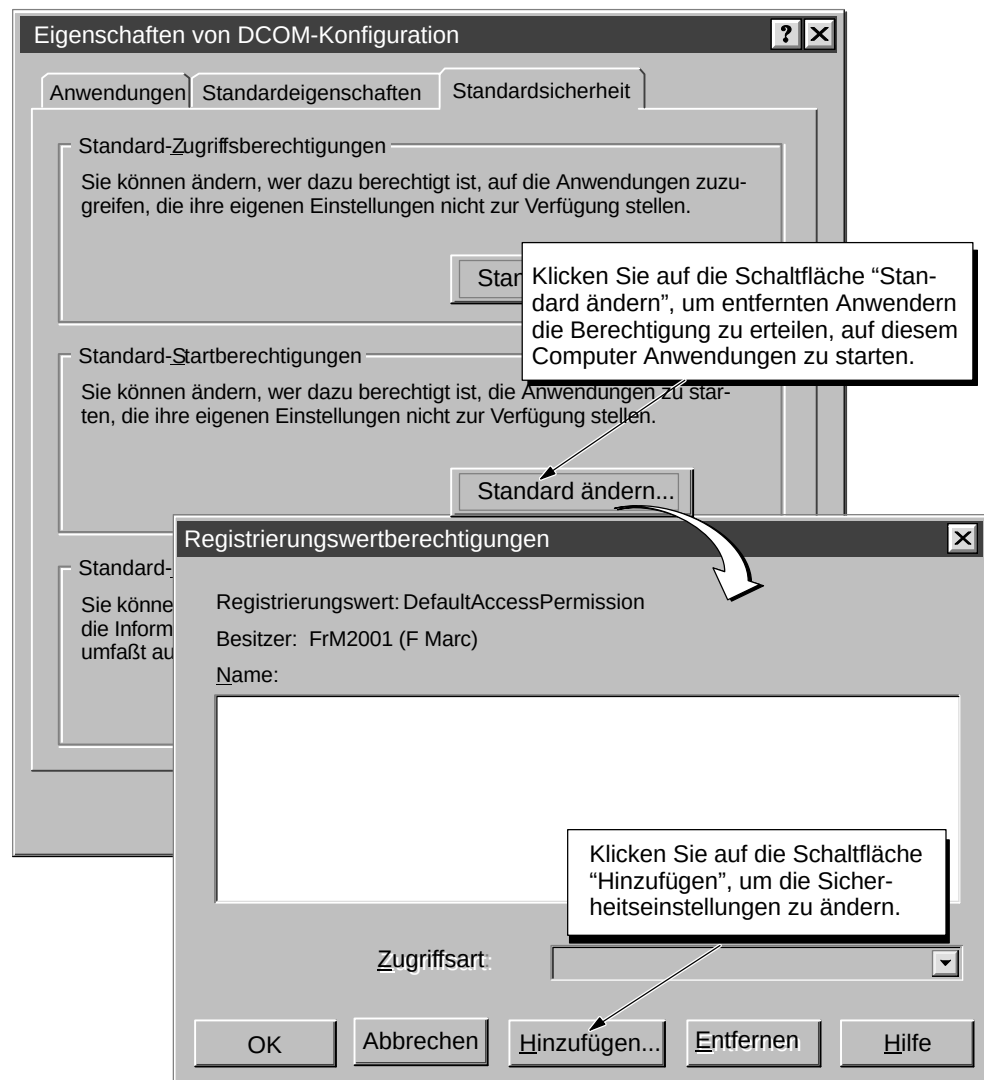


Bild E-17 Einrichten der Standard-Startberechtigungen für DCOM

3. Wählen Sie im Feld "Namen" im Dialogfeld "Benutzer und Gruppen hinzufügen" (Bild E-18) die Option "Everyone" (oder die gewünschte Benutzergruppe) und klicken Sie auf die Schaltfläche "Hinzufügen".

**Vorsicht**

Ist die Berechtigung für den Zugriff auf die Anwendungen erteilt, können andere Anwender (z.B. "Everyone") Programme starten und stoppen und auf Dateien in Ihrem Computer zugreifen. Bei nicht eingeschränktem Zugriff für alle Anwender im Netz kann es zu Problemen aufgrund unbeabsichtigter oder beabsichtigter Eingriffe kommen. Schränken Sie den Zugriff immer auf die Anwender ein, die mit den Anwendungen und Dateien auf dem Computer arbeiten müssen.

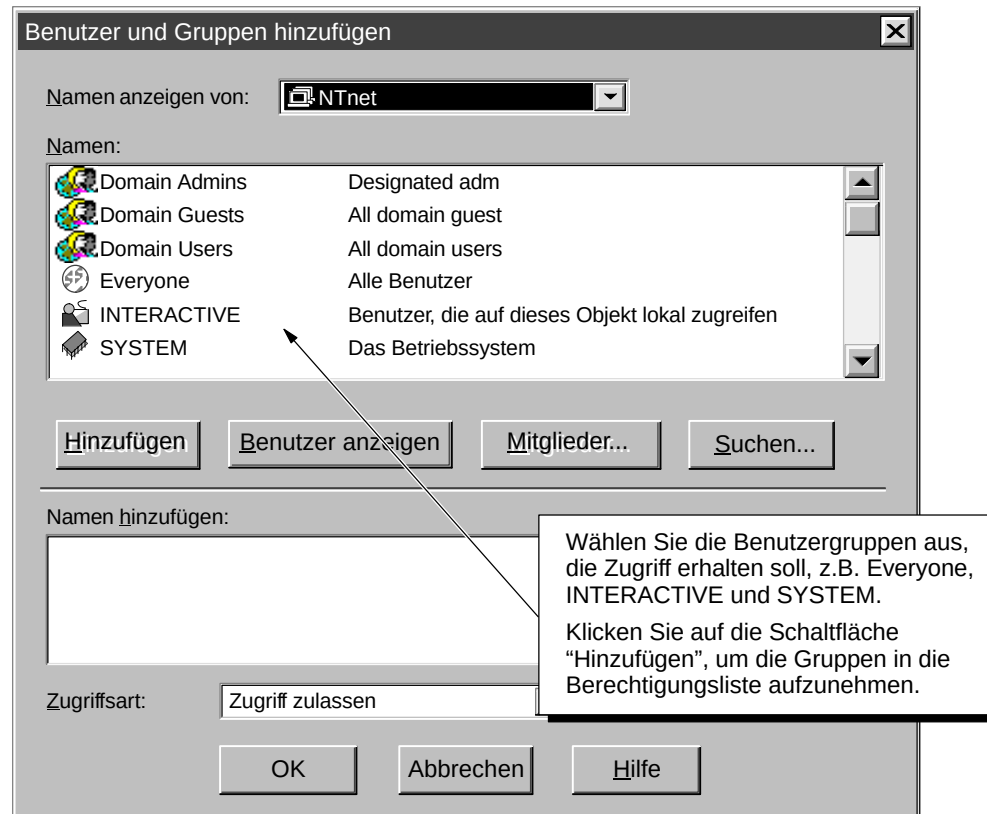


Bild E-18 Ändern der Startberechtigungen für Benutzer und Gruppen

4. Wählen Sie "INTERACTIVE" und klicken Sie auf die Schaltfläche "Hinzufügen".
5. Wählen Sie "SYSTEM" und klicken Sie auf die Schaltfläche "Hinzufügen".
6. Bestätigen Sie Ihre Eingaben mit "OK", die Änderungen werden daraufhin in das Dialogfeld "Registrierungswertberechtigungen" übernommen.
7. Wählen Sie im Dialogfeld "Registrierungswertberechtigungen" die Schaltfläche "OK", um die Änderungen an den Standardzugriffsberechtigungen einzugeben. Das Dialogfeld "Registrierungswertberechtigungen" wird geschlossen und zeigt wieder das Dialogfeld "Eigenschaften von DCOM-Konfiguration" an.

E.4 Fehlerbehebung

In diesem Abschnitt finden Sie Lösungen für einige der Probleme, die beim Einsatz von DCOM auftreten können. Ausführliche Informationen entnehmen Sie bitte dem Microsoft Online-Support (www.microsoft.com).

Probleme beim Lesen und Schreiben von Daten zwischen zwei Computern über DCOM

Situation: Sie arbeiten mit SIMATIC Computing auf dem Client-Computer (PC1) und sind über DCOM mit WinLC auf dem Server (PC2) verbunden. Sie möchten Daten zwischen den beiden Computern austauschen (lesen und schreiben), doch die Daten werden von WinLC auf PC2 nicht aktualisiert.

Möglicher Grund: PC1 wurde nicht so eingerichtet, daß PC2 aktualisierte Meldungen an PC1 senden kann.

Mögliche Abhilfe:

1. Starten Sie das Werkzeug für die DCOM-Konfiguration (dcomcnfg) auf PC1.
2. Öffnen Sie das Register "Sicherheit".
3. Klicken Sie auf die Schaltfläche "Standard ändern" für "Standard-Zugriffsberechtigungen", um das Dialogfeld "Registrierungswertberechtigungen" anzuzeigen.
4. Wählen Sie die Schaltfläche "Hinzufügen", um das Dialogfeld "Benutzer und Gruppen hinzufügen" aufzurufen und die Sicherheitseinstellungen für den Zugriff auf den Server zu ändern.
5. Wählen Sie im Feld "Namen" die Option "Everyone" und klicken Sie auf die Schaltfläche "Hinzufügen".
6. Bestätigen Sie Ihre Eingaben mit "OK", die Änderungen werden daraufhin in das Dialogfeld "Registrierungswertberechtigungen" übernommen.
7. Bestätigen Sie die Änderungen in den Standard-Zugriffsberechtigungen mit "OK".

Richtlinien zum Programmieren mit SIMATIC Computing



Kapitelübersicht

Das SIMATIC Data Control kann nicht nur mit anderen SIMATIC ActiveX Controls, sondern auch mit anderen Fremd-Controls oder anwenderspezifischen ActiveX Controls eingesetzt werden. Damit Sie mit einem anwenderspezifischen ActiveX Control arbeiten können, erfordert es das Data Control, daß das Control zumindest soviel Code zur Verfügung stellt, daß es auf Änderungen in der zugewiesenen Variable reagieren kann.

Wenn Sie Programme schreiben, die mit den SIMATIC Controls der Software SIMATIC Computing auf die Control Engine zugreifen, sollten Sie sich an die Richtlinien zum Programmieren halten, insbesondere an die Richtlinien bezüglich der Verwendung von Zeiten im Code.

Die Software SIMATIC Computing besteht aus einem Container (SoftContainer) für die SIMATIC Controls und andere ActiveX Controls. Sie können auch andere Container, z.B. Visual Basic, mit den SIMATIC Controls einsetzen. Damit Sie SIMATIC Controls in anderen Containern einsetzen können, muß der Container "extended Controls" unterstützen. Unterstützt der Container diese Funktionen nicht, müssen Sie Programmcode schreiben, der diese Funktionen erfüllt.

Kapitel	Beschreibung	Seite
F.1	Richtlinien für Fremd-Container	F-2
F.2	Richtlinien zum Programmieren	F-4
F.3	Richtlinien zum Erstellen von anwenderspezifischen ActiveX Controls	F-6
F.4	Einsetzen eines anwenderspezifischen ActiveX Control mit einem Data Control	F-7
F.5	Bekannte Probleme in SIMATIC Computing Version 3	F-10

F.1 Richtlinien für Fremd-Container

Damit das SIMATIC Data Control in einem Fremd-Container eingesetzt werden kann, muß der Container die Funktionen zum Suchen nach Eigenschaften für das Data Control unterstützen. Hierzu muß der Container die Funktionen für erweiterte Controls (extended controls: Definition von Microsoft für Container) unterstützen. Ein "extended Control" ist ein Teil-Control, das sich um ein anderes Control legt, um vom Container abhängige Eigenschaften, Methoden und Ereignisse zu unterstützen. (Ausführliche Informationen hierzu entnehmen Sie der Microsoft Online-Dokumentation zu Containern und erweiterten Controls).

Damit der Container die Funktionen für "extended Controls" bereitstellt, muß er folgende Methoden unterstützen:

- IOleClientSize::GetContainer
- IOleContainer::EnumObjects
- IOleControlSite::GetExtendedControl

Das "extended Control" des Containers muß auch die Eigenschaft Name unterstützen.

Der im Lieferumfang der Software SIMATIC Computing enthaltene SoftContainer unterstützt "extended Controls", genau wie Microsoft Visual Basic. Container von anderen Herstellern (z.B. Borland Delphi Version 3.0) unterstützen keine "extended Controls". Haben Sie Fragen dazu, ob Ihr Container die Funktionen für "extended Controls" unterstützt, wenden Sie sich bitte an den Siemens Customer Support.

Unterstützt der Container die "extended Controls" nicht, müssen Sie Programmcode schreiben, der diese Funktionen erfüllt. Vom Siemens Customer Support erhalten Sie Beispielcode, der die Funktionen für "extended Controls" ausführt.

OLE-Container

SIMATIC Computing ist ein offenes System, das mit OLE-Containern und Controls von verschiedenen Herstellern eingesetzt werden kann. Die SIMATIC Controls wurden mit folgenden Containern getestet:

- Microsoft Visual Basic 5.0
- Microsoft Visual Basic 6.0
- Microsoft Visual Basic for Applications (VBA) für Microsoft Office 97 Applikationen
- Microsoft Visual C++ von Microsoft Visual Studio 5.0 und 6.0
- WinAC SoftContainer, installiert mit der Software SIMATIC Computing

Einige andere Container von anderen Herstellern (z.B. Borland Delphi 3.0) unterstützen nicht alle erforderlichen ActiveX-Schnittstellen für die Funktionen zum Suchen nach Eigenschaften zwischen Data Control und anderen Controls. Für diese Container müssen Sie zusätzlichen Code in Ihrem Programm schreiben, um die Funktionen der "extended Controls" von Microsoft für Container zu unterstützen.

Weitere Informationen hierzu entnehmen Sie der Dokumentation (insbesondere der Auflistung der bekannten Probleme) des Fremd-Containers.

Wenn Sie die SIMATIC Controls in einem Fremd-Container einsetzen, lesen Sie sich bitte die Liste der bekannten Probleme des jeweiligen Containers durch.

Beispiel: Unter bestimmten Bedingungen kann es in Visual Basic 5.0 zu Ausnahmesituationen beim Schließen kommen. Dies beeinflusst den Betrieb von SIMATIC Computing nicht.

F.2 Richtlinien zum Programmieren

Folgende Richtlinien beziehen sich im speziellen auf Visual Basic, doch sind sie auch auf andere Programmiersprachen anwendbar.



Vorsicht

Wenn Sie die Zeitfunktion oder Haltepunkte in Unterprogrammen, die auf SIMATIC Computing zugreifen, verwenden, kann dies zu Fehlern führen, die evt. Ihren Computer bzw. die Anwendung zum Absturz bringen bzw. verriegeln können. Je nach Konfiguration kann die Anwendung dadurch die Kommunikation mit der Control Engine beenden. Dies kann unvorhersehbare Reaktionen im Prozeß auslösen, die zu Tod, schwerer Körperverletzung und/oder Sachschaden führen können.

Installieren Sie einen physikalischen NOT-AUS-Schaltkreis für die Maschine bzw. den Prozeß.

Verwenden von Timer-Funktionen in Ihrem Programm

Bei der Timer-Funktion in Visual Basic Version 5 kann ein Timer den in der gleichen Einweg-Bearbeitung befindlichen Code unterbrechen, wodurch Fehler mit schwerwiegenden Folgen auftreten können. Beachten Sie folgende Richtlinien, wenn Sie VB-Timer mit SIMATIC Computing einsetzen:

- Deaktivieren Sie die Timer immer im Unterprogramm Form_Unload. Andernfalls kann ein Timer ein Ereignis auslösen, während das VB-Programm geschlossen wird. Diese Bedingung kann einen Absturz bzw. eine Verriegelung Ihres Computers oder der Anwendung herbeiführen. Es kann auch sein, daß die Anwendung unsichtbar weiterläuft.
- Wenn Sie den Timer im Unterprogramm Form_Load starten, kann es sein, daß das Zeitereignis auftritt, bevor die anderen Objekte instanziiert wurden. Um sicherzustellen, daß die Objekte korrekt instanziiert werden, starten Sie einen Timer im Unterprogramm Form_Load immer mit einem großen Intervall (z.B. 1 oder 2 Sekunden). Nachfolgende Timer-Intervalle können kürzer sein.

Zugreifen auf kritische Daten mit einem eigenen Data Control

Die Leistungsfähigkeit Ihres Programms kann gesteigert werden, wenn Sie für den Zugriff auf kritische, sich häufig ändernde Daten ein eigenes SIMATIC Data Control einsetzen.

Aufheben der Verbindung zur Control Engine

Wenn Ihr Unterprogramm programmatisch auf das Data Control zugreift, müssen Sie (mit der Methode Disconnect) im Unterprogramm Form_Unload immer die Verbindung zur Control Engine trennen.

Außerdem müssen Sie, wenn Sie die Methode Connect in Ihrem Programm explizit aufrufen, die Eigenschaft AutoConnect für das Data Control deaktivieren. So stellen Sie sicher, daß das Data Control nicht unerwarteterweise mit der Control Engine verbunden wird.

Festlegen der Reihenfolge für AutoConnects für mehrere Data Controls

Wenn Sie in Ihrem Programm mehrere Data Controls verwenden, kann die Reihenfolge, in der die einzelnen Data Controls automatisch mit der Control Engine verbunden werden, nicht vorgegeben werden. Ist die Reihenfolge, in der die Data Controls mit der Control Engine verbunden werden, wichtig, müssen Sie die Eigenschaft AutoConnect für das Data Control deaktivieren und die Methoden Connect und Disconnect für die einzelnen Data Controls einsetzen.

F.3 Richtlinien zum Erstellen von anwenderspezifischen ActiveX Controls

Wenn Sie ein anwenderspezifisches ActiveX Control erstellen möchten, das mit dem SIMATIC Data Control eingesetzt werden kann, muß das anwenderspezifische Control über eine Eigenschaft verfügen, in die Daten geschrieben werden können. Das anwenderspezifische Control kann beispielsweise die Eigenschaft Value besitzen: ändert sich die Eigenschaft Value, reagiert das Control.

Lesen von Daten aus dem Data Control

Unterstützt der Container "extended Controls" (siehe Abschnitt F.1), sucht das Data Control automatisch das anwenderspezifische Control und seine Eigenschaften. Im Dialogfeld "Eigenschaften" des Data Control ordnen Sie einer Variablen in der Control Engine die Eigenschaft des anwenderspezifischen Control zu. (Informationen zum Zuordnen von Variablen zu Eigenschaften entnehmen Sie dem Abschnitt 5.4.) Immer wenn sich der Wert der Variablen in der Control Engine ändert, aktualisiert das Data Control den Wert der Eigenschaft für das anwenderspezifische Control.

Das anwenderspezifische Control sollte ein Unterprogramm für die Bearbeitung der vom Data Control geschriebenen Daten enthalten. Tabelle F-1 zeigt ein Beispiel für ein Unterprogramm für eine Eigenschaft (Value), die die vom Data Control geschriebenen Daten liest.

Schreiben von Daten in das Data Control

Damit das anwenderspezifische Control eine Änderung in die Variable in der Control Engine schreiben kann, müssen Sie ein Unterprogramm zum Bearbeiten der Änderungen in der Eigenschaft zur Verfügung stellen. Tabelle F-1 zeigt ein Beispiel für ein Unterprogramm zum Schreiben eines neuen Werts in das Data Control.

Tabelle F-1 Lesen und Schreiben eines geänderten Werts einer Eigenschaft

Visual Basic Code
<pre>Public Property Get Value() As Long Value = Object1.Value End Property</pre>
<pre>Public Property Let Value(ByVal New_Value As Long) Object1.Value() = New_Value PropertyChanged "Value" End Property</pre>
<pre>Private Sub Value_Change() PropertyChanged "Value" End Sub</pre>

F.4 Einsetzen eines anwenderspezifischen ActiveX Control mit einem Data Control

Sie können ein anwenderspezifisches ActiveX Control erstellen, das über das Data Control kommuniziert, um auf die Control Engine zugreifen zu können. Zum Erstellen dieser Beispielanwendung benötigen Sie folgendes:

- Microsoft Visual Basic 5 oder höher
- SIMATIC Data Control aus SIMATIC Computing
- Control Engine: z.B. WinLC oder eine Slot-PLC wie die CPU 416-2 DP ISA
- Beispielprogramm (siehe Abschnitt 1.1)
- STEP 7 (zum Laden des Programms in die Control Engine und zum Einschalten der Eingangsbits im Peripheriebereich des Beispielprogramms)

Sie können die Eingangsbits im Peripheriebereich des Beispielprogramms, das in der Control Engine ausgeführt wird, auch mit der Anwendung des Bedienpanel einschalten. Ausführliche Informationen zur Anwendung des Bedienpanel entnehmen Sie dem Abschnitt 1.2.

Erstellen eines anwenderspezifischen ActiveX Control zum Zugreifen auf die Control Engine

Gehen Sie folgendermaßen vor, um mit einem waagerechten Standard-Bildlauffeld in Visual Basic (HScrollBar Control) ein anwenderspezifisches ActiveX Control zu erstellen:

1. Öffnen Sie ein Visual Basic Projekt, um ein ActiveX Control zu erstellen: Wählen Sie den Menübefehl **Datei > Neues Projekt**, um das Dialogfeld "Neues Projekt" aufzurufen. Wählen Sie dann das Symbol "ActiveX Control" (**nicht** das Symbol "ActiveX EXE") und klicken Sie auf die Schaltfläche "Öffnen".
2. Hinzufügen eines Benutzersteuerelements zum Projekt: Wählen Sie den Menübefehl **Projekt > Benutzersteuerelement hinzufügen**, wählen Sie dann im Dialogfeld "Benutzersteuerelement hinzufügen" das Symbol "Benutzersteuerelement". Wenn Sie auf die Schaltfläche "Öffnen" klicken, wird das Benutzersteuerelement in das Projekt aufgenommen.
3. Wählen Sie das Control für ein waagerechtes Bildlauffeld (HScrollBar) in der Toolbox und fügen Sie es in das Arbeitsblatt UserControl1 ein.
4. Markieren Sie das Scrollbar Control. Wählen Sie im Fenster "Eigenschaften" die Eigenschaft Max für dieses Control (HScroll1) und geben Sie folgenden Wert ein:
255
5. Rufen Sie das Fenster für den Code von UserControl1 auf, indem Sie den Menübefehl **Ansicht > Code** wählen. Im Code-Fenster geben Sie das in Tabelle F-2 aufgeführte Programm ein.
6. Schließen Sie das Code- und das Objektfenster. Visual Basic nimmt dieses ActiveX Control (UserControl1) in die Toolbox auf.

Tabelle F-2 Beispielprogramm für ein ActiveX Control zum Einsatz in SIMATIC Computing

Visual Basic Code
<pre>Public Property Get Value() As Integer Value = HScroll1.Value End Property</pre>
<pre>Public Property Let Value (ByVal New_Value As Integer) HScroll1.Value = New_Value PropertyChanged "Value" End Property</pre>
<pre>Public Sub HScroll1_Change() Value = HScroll1.Value End Sub</pre>

Aufnehmen des anwenderspezifischen Control in ein Programm mit dem SIMATIC Data Control

1. Öffnen Sie ein neues Projekt in Visual Basic: Wählen Sie den Menübefehl **Datei ► Projekt ergänzen**, um das Dialogfeld "Projekt hinzufügen" aufzurufen. Wählen Sie dann das Symbol "Standard EXE" und klicken Sie auf die Schaltfläche "Öffnen". Visual Basic öffnet ein neues Projekt mit einem leeren Arbeitsblatt im Objektfenster.

Das Projektverzeichnis führt jetzt zwei Projekte auf: Project1 enthält UserControl1 und Project2 enthält Form1.
2. Wählen Sie das Symbol UserControl1 in der Toolbox und fügen Sie es in Form1 von Project2 ein.
3. Nehmen Sie das Siemens SIMATIC Data Control in die Toolbox auf. Ausführliche Informationen zum Aufnehmen von Controls in die VB Toolbox entnehmen Sie dem Abschnitt 1.1 und dem Bild 1-12.
4. Wählen Sie das Symbol Data Control in der Toolbox und fügen Sie es in Form1 von Project2 ein.
5. Markieren Sie das Data Control und klicken Sie mit der rechten Maustaste, um das Popup-Menü aufzurufen. Im Popup-Menü wählen Sie den Befehl **Eigenschaften**, um das Dialogfeld "Eigenschaften" des Data Control anzuzeigen.
6. Öffnen Sie im Dialogfeld "Eigenschaften" das Register "Connections". Klicken Sie auf das Symbol "+", um die Liste der Controls aufzuklappen.
7. Markieren Sie das UserControl1 Control und klicken Sie dessen Symbol "+" an, um die Liste der Eigenschaften aufzuklappen.
8. Wählen Sie die Eigenschaft Value und tragen Sie **AB0** in das Feld "Assigned Variable" ein (siehe Bild 1-13). Klicken Sie nacheinander auf die Schaltflächen "Übernehmen" und "OK", um die Daten einzugeben und das Dialogfeld "Eigenschaften" zu schließen.

Ausführen des Beispielprogramms

Speichern Sie das Programm, bevor Sie Visual Basic von der Betriebsart Design in RUN versetzen. Wird das Beispielprogramm ausgeführt, zeigt das von Ihnen erstellte Scrollbar Control den in AB0 abgelegten, sich ändernden Wert an.

Hinweis

Ist die Control Engine (z.B. WinLC oder eine Slot-PLC wie die CPU 416-2 DP ISA) nicht aktiv, kann das Data Control keine Verbindung aufbauen. Bevor Sie Visual Basic in den Betriebszustand RUN versetzen, vergewissern Sie sich, daß die Control Engine läuft.

Gehen Sie folgendermaßen vor, wenn Sie das Data Control für die Kommunikation mit der Control Engine und für die Ausführung des Beispielprogramms konfigurieren möchten.

1. Öffnen Sie das Register "Engine", um die Control Engine einzurichten (siehe Bild F-1).
2. Wählen Sie die Option "Direct Connect" und geben Sie entweder **wcS7=3** (Groß-/Kleinschreibung beachten) für eine Slot-PLC wie die CPU 416-2 DP ISA oder **WinLC** als Control Engine an. (Die Zeichenfolgen für andere Control Engines entnehmen Sie dem Anhang G.) Wählen Sie die Schaltfläche "Übernehmen", um die Daten zu übernehmen und bestätigen Sie mit "OK", um das Dialogfeld zu schließen.
3. Versetzen Sie Visual Basic von der Betriebsart Design in RUN, um das Beispielprogramm auszuführen.

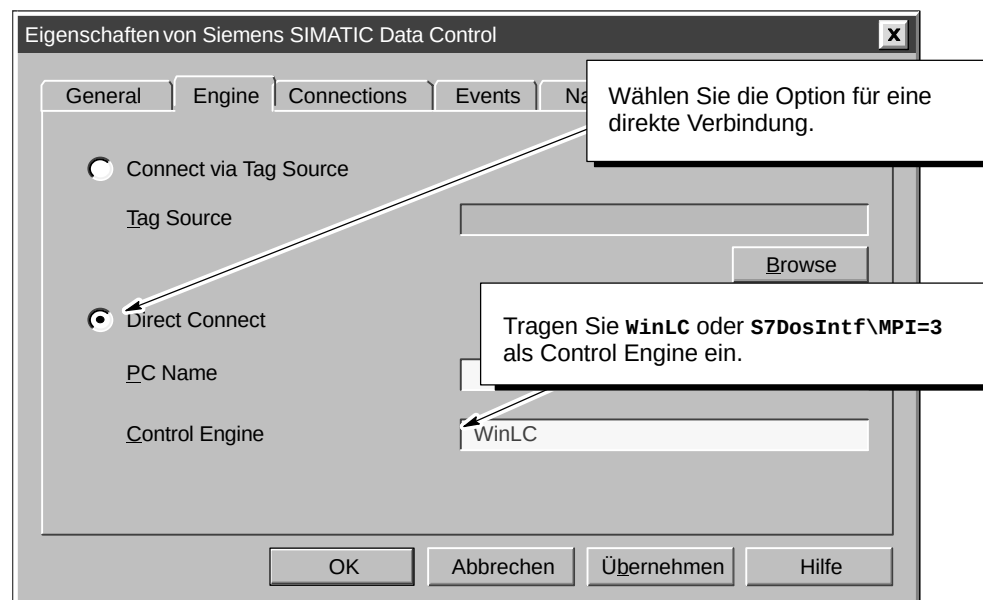


Bild F-1 Verbinden mit der Control Engine (Beispiel: Scrollbar Control)

F.5 Bekannte Probleme in SIMATIC Computing Version 3

Schreiben in die Eingänge der Peripherie

Bei Control Engines (Automatisierungssystemen) können Sie nicht in den Speicherbereich der Peripherie der Eingänge (PE) schreiben. Sie können zwar in SIMATIC Computing Daten im Speicherbereich PE lesen, doch Sie können nicht in den Speicherbereich PE schreiben.

Lesen der Ausgänge der Peripherie

Es ist nicht zu empfehlen, daß Ihr Programm aus dem Speicherbereich der Peripherie der Ausgänge (PA) liest.

Während Sie bei S7 Control Engines (Automatisierungssystemen) üblicherweise in den Peripheriebereich der Ausgänge (Speicherbereich PA) schreiben können (entweder aus STEP 7 oder aus Ihrem Programm), können Sie bei S7 Control Engines nicht den Speicherbereich PA lesen. In SIMATIC Computing kann jedoch Ihr Programm die Werte des Peripheriebereichs der Ausgänge lesen. Achten Sie darauf, daß die aus dem Speicherbereich PA gelesenen Werte nicht immer genau die zu erwartenden Werte wiedergeben:

- Befindet sich die Control Engine im Betriebszustand RUN: Die Werte des Peripheriebereichs der Ausgänge (PA) sind üblicherweise korrekt. Diese Werte können jedoch falsch sein, wenn das entsprechende E/A-Modul fehlerhaft ist oder ganz fehlt.
- Befindet sich die Control Engine im Betriebszustand STOP: Die Werte des Peripheriebereichs der Ausgänge (PA) zeigen die konfigurierten Ersatzwerte für den sicheren Zustand an.

In SIMATIC Computing können Sie keine Daten in den Peripheriebereich der Ausgänge (PA) schreiben, wenn sich die Control Engine im Betriebszustand STOP befindet.

Erkennen einer verlorenen MPI-Verbindung

Das Data Control erkennt den Verlust einer MPI-Verbindung nicht. Gehen Sie folgendermaßen vor, um eine verlorene MPI-Verbindung zu erkennen:

1. Fügen Sie Ihrem Programm eine Zeit hinzu.
2. Lesen Sie in bestimmten Abständen (z.B. 1 s) mit der Methode ReadVariable des Data Control eine bestimmte Variable (z.B. MB0) aus.
3. Erhalten Sie eine Fehlermeldung, daß die Methode ReadVariable fehlgeschlagen ist, haben Sie die MPI-Verbindung verloren. Ihr Programm kann dann auf die verlorene Verbindung reagieren.

Handhabung von OPC-Fehlern in Visual Basic

OPC-Methoden geben Fehlercodes an HRESULT (Variable Long im Hexadezimalformat) aus. In Visual C werden Fehlerbedingungen mit HRESULT bearbeitet. In Visual Basic wird die Fehlerbehebung in das VB-Fehlerobjekt (ERR) geschrieben. Sie müssen in Ihrem VB-Programm Code ergänzen, um auf die Fehlercodes vom OPC-Interface zugreifen zu können.

Umwandeln von STEP 7 Time of Day (TOD) in Visual Basic vbDate

Es tritt ein Fehler auf, wenn Sie eine STEP 7 Variable vom Datentyp TOD lesen und den Wert in das Format Visual Basic vbDate umwandeln. Wenn Sie eine Variable während der letzten halben Sekunde des Tages lesen (zwischen 23h59m59s500ms und 23h59m59s999ms), zeigt vbDate das folgende Datum für die Variable an: 31.12.1899 (31. Dezember 1899).

Die Zeitwerte sind korrekt. Diese Unregelmäßigkeit tritt nur während der letzten halben Sekunde eines Tages auf und steht nicht in Zusammenhang mit der Jahr-2000-Fähigkeit ("Y2K").

Fehlercodes in SIMATIC Computing Version 2.0.1 und 3.0 sind nicht kompatibel mit denen in Version 1.2 und Version 1.1



Warnung

Wenn Sie Fehlerbedingungen in Ihrem Programm nicht richtig beheben, können Sie unvorhersehbare Reaktionen im Prozeß auslösen, die zu Tod, schwerer Körperverletzung und/oder Sachschaden führen können.

Stellen Sie sicher, daß Programme zur Behebung von Fehlern aus früheren Versionen von SIMATIC Computing (Versionen 1.1 oder 1.2) für die Fehlercodes der Versionen 2.0.1 und 3.0 von SIMATIC Computing aktualisiert wurden. Gehen Sie vorsichtig vor, ändern Sie keine Daten und gewähren Sie unberechtigten Personen keinen Zugriff auf Daten, die unerwartete Funktionsweise der gesteuerten Geräte hervorrufen können. Installieren Sie einen physikalischen NOT-AUS-Schaltkreis für die Maschine bzw. den Prozeß.

Viele Fehlercodes von SIMATIC Computing wurden für die Versionen 3.0 und 2.0.1 geändert. Programme, die für frühere Versionen von SIMATIC Computing geschrieben wurden, reagieren nicht unbedingt richtig auf die Fehlerbedingung und müssen aktualisiert werden.

Verbinden von mehreren Control Engines über ein einziges Data Control

Das Verbinden von mehreren Control Engines über ein einziges Data Control bewirkt, daß der Container nicht mehr auf Wechsel vom Betriebszustand Design in den Betriebszustand RUN reagiert. Wenn Sie eine Symboldatei verwenden, die mehrere Control Engines enthält, setzen Sie für jede Control Engine ein eigenes Data Control ein. Die Data Controls können dieselbe Symboldatei nutzen, doch jedes Data Control darf nur Controls mit Symbolen einer Control Engine verbinden. Dieses Problem wird mit dem nächsten Ausgabestand bzw. Service Pack behoben sein.

Verbinden der Control Engine über Control Arrays in VB

Wenn Sie eine Verbindungstabelle programmatisch erstellen (mit der Methode `ConnectObject` im Code in Ihrem Visual Basic-Programm zum Verbinden von Objekten) und dann mit Hilfe dieser Verbindungstabelle Elemente eines Control Array mit der Control Engine verbinden, werden Werte, die von Elementen des Control Array geändert werden, nicht automatisch in die Control Engine geschrieben. Während das Data Control automatisch die von der Control Engine vorgenommenen Änderungen aktualisiert (indem die geänderten Werte automatisch in die Elemente des Control Array gelesen werden), werden die geänderten Werte (durch das Control Array) jedoch nicht automatisch in die Control Engine geschrieben.

- Wenn Sie möchten, daß mit dem Control Array vorgenommene Änderungen über die Option "Automatic Update" des Data Control automatisch in die Control Engine geschrieben werden, rufen Sie das Dialogfeld "Eigenschaften" des Data Control auf und erstellen die Verbindungen für das Control Array (anstatt mit der Methode `ConnectObject` Code in Ihr VB-Programm einzufügen, um die Verbindungen herzustellen).

Wenn Sie das Dialogfeld "Eigenschaften" des Data Control einsetzen, um die Elemente des Control Array zu suchen und in der Control Engine Variablen zuzuordnen, werden mit den Elementen des Control Array durchgeführte Änderungen automatisch in die Control Engine geschrieben. (Achten Sie darauf, daß die Option "Automatic Update" für das Data Control aktiviert ist.)

- Wenn Sie nicht möchten, daß mit dem Control Array vorgenommene Änderungen automatisch in die Control Engine geschrieben werden, können Sie Code in Ihr VB-Programm implementieren (z.B. in den Code für einen Button Control), um den geänderten Wert in das Control zu schreiben, wobei mit Hilfe der Methode `WriteVariable` oder `WriteMultiVariables` der Wert manuell in der Control Engine aktualisiert wird.

Unterschiede in den Fehlercodes bei unterschiedlichen Automatisierungssystemen

Das Data Control gibt für unterschiedliche Automatisierungssysteme unterschiedliche Fehlercodes aus (siehe Tabelle F-3).

Tabelle F-3 Fehlercodes

Fehlerbedingung	WinLC	CPU 416-2 DP ISA und andere S7-Automatisierungssysteme
In schreibgeschützten DB schreiben	0xC0040006	0xC0040007
Falscher Variablenname	0xC0040008	0xC0040007
Außerhalb des Speicherbereichs (z.B. mb40000)	0xC0040007	0 x 80070057

Unterschiede zwischen WinLC und einer Slot-PLC

Zwischen WinLC (WinAC Basis) und einer Slot-PLC wie der CPU 416-2 DP ISA (WinAC Pro) gibt es folgende Unterschiede:

- WinLC unterstützt Arrays der folgenden Datentypen: BOOL, CHAR und STRING. MPI-Teilnehmer (wie die CPU 416-2 DP ISA und andere S7-Controller) unterstützen Arrays dieser Datentypen nicht.
- Obwohl ein solches Vorgehen nicht empfohlen wird, können Sie in WinLC den Peripheriebereich der Ausgänge (PA) lesen. Andere S7-Controller (wie die CPU 416-2 DP ISA) können den Peripheriebereich der Ausgänge (PA) nicht lesen.

Fehlerbehebung: Verzögerte Reaktionen von Software mit Einsatz von COM

Ihre DCOM-Konfiguration kann lokale COM-Operationen beeinflussen. Stellen Sie beispielsweise für den "Standard-Authentication-Level" "Kein" (anstelle von "Verbinden") ein, kann dies die Verbindung zu Software-Anwendungen um bis zu 6 Minuten verzögern, wenn das Betriebssystem Windows NT die Sicherheitsprüfungen ausführt. Dies wirkt sich nicht nur auf WinAC-Komponenten (z.B. SIMATIC Computing oder WinLC), sondern auch auf andere Software-Anwendungen, die Kommunikation einsetzen (wie Microsoft Word), aus.

Wenn Sie Ihren Computer für DCOM konfigurieren, richten Sie sich bitte nach den in Anhang E angegebenen Einträgen.

Zeichenfolgen für Control Engines

Übersicht

Zeichenfolgen der Control Engines werden im Data Control, im Symboldatei-Konfigurator und im OPC und Diagnostic Buffer Control verwendet. Die Zeichenfolge der Control Engine kennzeichnet den Speicherplatz der Control Engine des STEP 7 Projekts.

Hinweis

Von einem PC können Sie nur auf eine Slot-PLC oder ein S7-Netz zur Zeit zugreifen.

Die Syntax der Zeichenfolge der Control Engine richtet sich nach der Art der Control Engine und dem verwendeten Netz. In der Formel **wcS7=xx,a,b** ist wcS7 beispielsweise eine S7-CPU in einem S7-Netz; xx, a und b definieren die Anordnung der CPU (xx ist die Teilnehmeradresse der CPU, a ist die Nummer des Baugruppenträgers und b ist die Nummer des Steckplatzes).

Hinweis

Die Nummer für Baugruppenträger und Steckplatz sind optional, doch ist die Baugruppenträgernummer angegeben, muß auch die Nummer für den Steckplatz angegeben werden. Ist nichts angegeben, wird auf die Baugruppe bzw. CPU direkt zugegriffen.

Einstellungen der Control Engine für WinLC

Zum Zugreifen auf WinLC geben Sie **WinLC** für die Eigenschaft "Control Engine" an.

Zum Zugreifen auf WinLC über TCP/IP LAN müssen Sie den Namen des PCs, auf dem WinLC installiert ist, als Eigenschaft "PCName" (Feld "Computername" im Symboldatei-Konfigurator) in einem Wort ohne Leerzeichen (z.B. PC_2) angeben. Sie brauchen keine Einstellungen in der Anwendung "PG/PC-Schnittstelle einstellen" für WinLC einzurichten.

Wie Sie auf WinLC über ein S7-Netz zugreifen, erfahren Sie im Abschnitt "Einstellungen der Control Engine in einem SIMATIC Netz" weiter unten. In einem S7-Netz müssen Sie für die Adressen von Baugruppenträger und Steckplatz für WinLC die Werte 0,2 angeben.

Einstellungen der Control Engine für eine Slot-PLC

Zum Zugreifen auf eine Slot-PLC wie die CPU 416-2 DP ISA geben Sie **wcS7=3** für die Eigenschaft "Control Engine" an.

Zum Einstellen der SIMATIC Computing Schnittstelleneinstellungen für die CPU 416-2 DP ISA auf dem PC, auf dem die CPU installiert ist, müssen Sie in "PG/PC-Schnittstelle einstellen" folgende Einstellungen einrichten:

COMPUTING-->CPU 416-2 DP ISA (lokal)

Zum Zugreifen auf die CPU 416-2 DP ISA über TCP/IP LAN müssen Sie den Namen des PCs, auf dem die CPU installiert ist, als Eigenschaft "PCName" (Feld "Computername" im Symboldatei-Konfigurator) in einem Wort ohne Leerzeichen angeben.

Wie Sie auf eine Slot-PLC über ein S7-Netz zugreifen, erfahren Sie im Abschnitt "Einstellungen der Control Engine in einem SIMATIC Netz" weiter unten. In einem S7-Netz müssen Sie für die Adressen von Baugruppenträger und Steckplatz für die Slot-PLC die Werte 0,3 angeben.

Einstellungen der Control Engine in einem SIMATIC Netz

Zum Zugreifen auf ein S7-System in einem MPI- oder PROFIBUS-Netz verwenden Sie die Formel **wcS7=xx,a,b**. Hierbei steht xx für die Teilnehmeradresse, a ist die Nummer des Baugruppenträgers und b ist die Nummer des Steckplatzes. (Der längere Legacy-String **S7DosIntfMPI=x,a,b** wird auch unterstützt.) Beachten Sie folgendes:

- Die Teilnehmeradresse ist immer dezimal.
- Die Nummer des Baugruppenträgers ist 0 bis 7 dezimal.
- Die Nummer des Steckplatzes ist 0 bis 31 dezimal.

Zum Zugreifen auf ein S7-System in einem TCP/IP LAN verwenden Sie die Formel **wcIP=xxx.xxx.xxx.xxx,a,b**. Hierbei steht xxx.xxx.xxx.xxx für die TCP/IP-Adresse, a ist die Nummer des Baugruppenträgers und b ist die Nummer des Steckplatzes. Beachten Sie folgendes:

- Die TCP/IP-Adresse besteht aus vier Dezimalzahlen, die durch Punkte getrennt sind.
- Die Nummer des Baugruppenträgers ist 0 bis 7 dezimal.
- Die Nummer des Steckplatzes ist 0 bis 31 dezimal.

Beindet sich die Slot-PLC auf demselben Computer wie die Software SIMATIC Computing, stellen Sie die PG/PC-Schnittstelle folgendermaßen ein:

COMPUTING--><Ethernet-Kartenname>(TCP/IP).

Zum Zugreifen auf ein S7-System in einem Industrial Ethernet verwenden Sie die Formel $wcMAC=xx.xx.xx.xx.xx.xx,a,b$. Hierbei steht $xx.xx.xx.xx.xx.xx$ für die MAC-Adresse, a ist die Nummer des Baugruppenträgers und b ist die Nummer des Steckplatzes. Beachten Sie folgendes:

- Die TCP/IP-Adresse besteht aus sechs Dezimalzahlen, die durch Punkte getrennt sind.
- Die Nummer des Baugruppenträgers ist 0 bis 7 dezimal.
- Die Nummer des Steckplatzes ist 0 bis 31 dezimal.

Befindet sich die Slot-PLC auf demselben Computer wie die Software SIMATIC Computing, stellen Sie die PG/PC-Schnittstelle folgendermaßen ein:

COMPUTING--><Ethernet-Kartenname>(ISO Transport).

Wenn die Kommunikationskarte auf einem anderen PC installiert ist als SIMATIC SIMATIC Computing, geben Sie den Namen des PCs als Eigenschaft "PCName" (Feld "Computername" im Symboldatei-Konfigurator) in einem Wort ohne Leerzeichen an (z.B: PC_2).

Beispiel: In diesem Beispiel wird auf eine S7-315 CPU (immer in Steckplatz 2) mit der Teilnehmeradresse 5 zugegriffen.

- Zum Zugreifen auf die CPU über ein S7-Netz lautet die Zeichenfolge für die Control Engine folgendermaßen: **wcS7=5,0,2.**
- Zum Zugreifen auf die CPU über ein Industrial Ethernet lautet die Zeichenfolge für die Control Engine folgendermaßen: **wcMAC=a.b0.12.ff.3.2d,0,2.**
- Zum Zugreifen auf die CPU über TCP/IP LAN lautet die Zeichenfolge für die Control Engine folgendermaßen: **wcIP=0.0.255.255,0,2.**

Hinweis

In einem S7-Netz lautet die Adresse von Baugruppenträger und Steckplatz für WinLC immer 0,2. Die Adresse von Baugruppenträger und Steckplatz für eine Slot-PLC wie die CPU 4116-2 DP ISA lautet 0,3.

Index

A

- Absolute Adressen
 - Ersetzen von Symbolen, 5-11
 - Symboldatei, 9-9
- ActiveX Controls
 - Anlegen eines Prozeß-Formulars, 8-4 – 8-6
 - anwenderspezifische Controls, F-7 – F-10
 - Beispielprogramm
 - Bedienpanel, 1-4 – 1-10
 - Microsoft Excel, 1-15 – 1-19
 - SoftContainer, 1-19 – 1-25
 - STEP 7 Programm, 1-3
 - weitere Controls (VBS scrollbar), 1-12 – 1-15
- Button, 6-4 – 6-8
- Button Control
 - Beschreibung, 6-4
 - Schaltfläche in der Funktionsleiste, 6-4
- Data
 - Beschreibung, 5-1
 - Schaltfläche in der Funktionsleiste, 5-1
- Data Control
 - Ereignisse, 5-30
 - Fehlercodes, 5-30
- Edit Control
 - Beschreibung, 6-11
 - Schaltfläche in der Funktionsleiste, 6-11
- Eigenschaften
 - Button Control, 6-2 – 6-6
 - Data Control, 5-29 – 5-30
- Erstellen, F-6 – F-10
- Filtern von Eigenschaften, 5-13
- Gemeinsames Nutzen von Daten in mehreren Applikationen, 2-3, 2-4 – 2-6
- Label, 6-21 – 6-25
- Label Control
 - Beschreibung, 6-21
 - Schaltfläche in der Funktionsleiste, 6-21
- mit SIMATIC Controls, 5-1 – 5-12
- Slider Control, Beschreibung, 6-27
- SoftContainer
 - Betriebszustand, 8-6 – 8-8
 - Übersicht, 8-2 – 8-4
- Verbinden mit der Control Engine, 5-9
- Ändern der Sprache, D-5

- Anforderungen an das Betriebssystem, 2-3
- Anforderungen an den Computer, 2-3
- Anwenderspezifische Ereignisse, 5-15
- Ausgänge, Speicherbereiche A und PA von S7-Automatisierungssystemen, A-2
- Ausgänge von S7-Automatisierungssystemen
 - Zugreifen auf den Speicherbereich PA, A-2
- Auswirkungen des Zyklus auf Eingänge und Ausgänge, A-2
- AUTHORS.EXE
 - Autorisierung für SIMATIC Computing, 3-2 – 3-5
 - Betrieb ohne Autorisierung, 3-4
 - Deinstallieren der Autorisierung für SIMATIC Computing, 3-4
 - Installation (SIMATIC Computing), 3-5
 - Übertragen der Autorisierung für SIMATIC Computing, 3-4

B

- Beispiel
 - Beispielprogramm, 1-2
 - Lesen/Schreiben von booleschen Daten, 5-28
 - Lesen/Schreiben von Daten, 5-23
 - Programm für eine Ereignistabelle, 5-18
 - Programm zum Erstellen einer Verbindungstabelle, 5-17
 - Programm zum Reagieren auf Ereignisse, 5-19 – 5-22
- Beispiel in Visual Basic, Lesen/Schreiben von Daten, 5-23
- Beispiele
 - anwenderspezifische ActiveX Controls, F-7 – F-10
 - Bedienpanel, 1-4 – 1-10
 - Lesen/Schreiben mit dem Data Control, F-6
 - Microsoft Excel, 1-15 – 1-19
 - SoftContainer, 1-19 – 1-25
 - STEP 7 Programm, 1-3
 - weitere Controls (VBS scrollbar), 1-12 – 1-15
- Beispielprogramme
 - anwenderspezifisches ActiveX Control, F-7 – F-10

Beispielprogramme
 Bedienpanel, 1-4 – 1-10
 Lesen/Schreiben mit dem Data Control, F-6
 Microsoft Excel, 1-15 – 1-19
 SoftContainer, 1-19 – 1-25
 STEP 7 Programm, 1-3
 weitere Controls (VBScrollbar), 1-12 – 1-15
 Betriebszustand Design, SoftContainer, 8-6 – 8-8
 Betriebszustand RUN, SoftContainer, 8-6 – 8-8
 Darstellung des Data Control, 5-1
 BSTR, A-9
 Button Control, 2-4
 Beschreibung, 2-4, 6-4
 Eigenschaften, 6-2 – 6-6
 Alignment, B-2
 Appearance, B-3
 BorderStyle, B-6
 Enabled, B-19
 FalseCaption, B-21
 FalseColor, B-21
 FalsePicture, B-22
 Font, B-22
 ForeColor, B-23
 Locked, B-26
 PushButton, B-32
 StretchMode, B-38
 Style, B-39
 TrueCaption, B-40, B-42
 TrueColor, B-41
 Value, B-43
 Eigenschaften und Methoden, 6-9
 Ereignisse, 6-10
 Change, C-1
 Click, C-1
 Error, C-2
 KeyDown, C-3
 KeyPress, C-5
 KeyUp, C-5
 MouseDown, C-6
 MouseMove, C-7
 MouseUp, C-8
 Methoden, AboutBox, B-1
 Schaltfläche in der Funktionsleiste, 6-4

C

Client-Anwendung (OPC), Verbinden mit
 SIMATIC Computing, 2-6 – 2-7
 Client-Applikation (OPC), 2-3, 2-6 – 2-8
 Server-Name, 2-7
 Server-Schnittstellen, 2-7
 Client-Konfiguration (DCOM), E-14 – E-19

Component Object Model (COM)
 Client und Server, E-1 – E-20
 lokal und entfernt, 4-1 – 4-7
 Control Engine
 Ändern des Namens in der Symboldatei, 9-13
 Auswirkungen des Zyklus auf Eingänge und Ausgänge, A-2
 Einrichten für entfernten Zugriff, 9-12
 Einrichten für lokalen Zugriff, 9-11
 Einstellen der lokalen Control Engine, D-6
 Einstellen für das Data Control, 5-4
 Entfernt, 4-5
 lokal, 4-4
 mehrere, 4-6
 OPC Controls, 2-6 – 2-8
 OPC-Verbindung, D-2
 OPC-Zugriff, 2-6
 SIMATIC Controls, 5-1 – 5-12
 Button, 6-4 – 6-8
 Edit, 6-11 – 6-22
 Label, 6-21 – 6-25
 Slider, 6-27 – 6-29
 Speicherbereiche von S7-Automatisierungssystemen, A-2
 Symboldateien, 9-1
 Verbinden mit SIMATIC Controls, 5-2
 Verbinden über DCOM, 5-6, 9-2
 Verbinden von ActiveX Controls, 5-9
 Zugreifen auf Daten, 4-2 – 4-6
 Zugriff, 2-3 – 2-6
 Controller. *siehe* Control Engine
 CPU
 PC-Anforderungen, 2-3
 Speicherbereiche von S7-Automatisierungssystemen, A-2
 CPU 416-2 DP ISA. *siehe* Control Engine

D

Data, 2-4
 ActiveX Controls, Data, 5-1 – 5-13
 OPC Controls, 2-6 – 2-8
 SIMATIC Controls, 5-1 – 5-12
 Button, 6-4 – 6-8
 Edit, 6-11 – 6-22
 Label, 6-21 – 6-25
 Slider, 6-27 – 6-29
 Data Control, 2-3
 anwenderspezifische ActiveX Controls, F-6 – F-10
 Beispielprogramm
 Bedienpanel, 1-4 – 1-10

Data Control

- Microsoft Excel, 1-15 – 1-19
- SoftContainer, 1-19 – 1-25
- STEP 7 Programm, 1-3
- weitere Controls (VBScrollbar),
1-12 – 1-15

Beschreibung, 2-4, 5-1

Container, F-2

Eigenschaften, 5-29 – 5-30

- Activated, B-1
- AutoConnect, B-3
- AutoConnectTimeout, B-4
- ControlEngine, B-10
- DefaultDeadband, B-12
- DefaultUpdateRate, B-13
- Methode WriteMultiVariables, B-44
- Methode WriteVariable, B-45
- MultipleEngines, B-27
- PCName, B-28
- ReadMultiVariables, B-33
- ReadVariable, B-33
- ShowErrorBoxes, B-36
- TagSource, B-39

Einrichten der Verbindungseigenschaften,
5-3

Einrichten für eine Control Engine, 5-6

Einstellen der Control Engine, 5-4

Ereignisse, 5-30

- ConnectionError, C-1
- ValueChanged, C-9

Ergänzen eines Ereignisses, 5-15

Fehlercodes, 5-30 – 5-32

Konfigurieren von mehreren Control
Engines, 9-2 – 9-4

Methoden, 5-29

- Connect, B-7
- ConnectName, B-8
- ConnectObject, B-9
- Disconnect, B-14
- PropertyChangedName, B-30
- PropertyChangedObject, B-31

Schaltfläche in der Funktionsleiste, 5-1

SoftContainer

- Betriebszustand, 8-6 – 8-8
- Prozeß-Formular, 8-4 – 8-6
- Übersicht, 8-2 – 8-4

Verbindungen, 5-9

Verbindungstabelle, 5-16

Daten

- ActiveX Controls, Button, 6-2 – 6-6
- gemeinsames Nutzen von Daten in mehre-
ren Applikationen, 2-3 – 2-5

Daten

- Speicherbereiche von S7-Automatisie-
rungssystemen, A-2
- Zugreifen mit SIMATIC Computing,
2-3 – 2-5

Datenbanken, gemeinsames Nutzen von Da-
ten über OPC, 2-6

Datentyp ARRAY, A-6 – A-8

Datentyp CHAR, A-9

- BSTR (Visual Basic), A-9

Datentyp DATE, A-7

Datentyp S5TIME, A-8

Datentyp STRING, BSTR (Visual Basic), A-9

Datentyp TIME, A-10

Datentyp TIME_OF_DAY, A-10

DBuffer Control

- Beschreibung, 2-3, 7-1
- Eigenschaften, 7-7 – 7-8
- Konfigurieren, 7-4
- Methoden, 7-7

DBuffer control, 7-2 – 7-6

DCOM

- Client und Server, E-1 – E-20
- Client-Konfiguration, E-14 – E-19
- Fehlerbehebung, E-20
- Konfigurationseditor, E-4, E-14
- lokal und entfernt, 4-1 – 4-7
- Server-Konfiguration, E-4 – E-13

Deinstallieren, 3-6

Deinstallieren der Autorisierung für SIMATIC
Computing, 3-3 – 3-5

Deinstallieren der Autorisierung für WinLC,
Richtlinien, 3-3

Deinstallieren der Software SIMATIC
Computing, 3-6

Deutsch, Ändern, D-5

Dezentrale Anwendungen (DCOM)

- Einrichten von Server und Client,
E-1 – E-20

- lokal und entfernt, 4-1 – 4-7

Diagnosepuffer, DBuffer Control, 7-2

Distributed Component Object Model (DCOM)
siehe auch DCOM

- Client und Server, E-1 – E-20
- lokal und entfernt, 4-1 – 4-7

E

Edit Control, 2-4

- Beschreibung, 2-4, 6-11
- Eigenschaften

Edit Control

- Alignment, B-2
- Appearance, B-3
- BackColor, B-5
- BorderStyle, B-6
- DisplayValue, B-18
- Enabled, B-19
- Factor, B-20
- Font, B-22
- ForeColor, B-23
- Locked, B-26
- Max und Min, B-26
- Offset, B-27
- Precision, B-30
- RawMax, B-32
- RawMin, B-32
- ScaleMode, B-34
- Text, B-40
- Value, B-43
- WriteMode, B-43
- WriteNow Methode, B-44
- ZeroPad, B-46

Eigenschaften und Methoden, 6-18

Ereignisse, 6-19

- Change, C-1
- Click, C-1
- DbClick, C-2
- Error, C-2
- KeyDown, C-3
- KeyPress, C-5
- KeyUp, C-5
- MouseDown, C-6
- MouseMove, C-7
- MouseUp, C-8

Fehlercodes, 6-20

Methoden, AboutBox, B-1

Schaltfläche in der Funktionsleiste, 6-11

- Eigenschaft Activated, B-1
- Eigenschaft Alignment, B-2
- Eigenschaft Appearance, B-3
- Eigenschaft AutoConnect, B-3
- Eigenschaft AutoConnectTimeout, B-4
- Eigenschaft BackColor, B-5
- Eigenschaft bDiagBuffOK, B-5
- Eigenschaft bEngineConnected, B-6
- Eigenschaft BorderStyle, B-6
- Eigenschaft ControlEngine, B-10
- Eigenschaft DataType, B-11
- Eigenschaft DefaultDeadband, B-12
- Eigenschaft DefaultUpdateRate, B-13
- Eigenschaft Direction, B-14
- Eigenschaft DisplayFormatButtons, B-15
- Eigenschaft DisplayHelpButton, B-15

- Eigenschaft DisplayHelpOnEventButton, B-16
- Eigenschaft DisplayLowerPanel, B-17
- Eigenschaft DisplayUpdateButton, B-17
- Eigenschaft DisplayUpperPanel, B-18
- Eigenschaft DisplayValue, B-18
- Eigenschaft Enabled, B-19
- Eigenschaft EnableSort, B-19
- Eigenschaft Factor, B-20
- Eigenschaft FalseCaption, B-21
- Eigenschaft FalseColor, B-21
- Eigenschaft FalsePicture, B-22
- Eigenschaft Font, B-22
- Eigenschaft ForeColor, B-23
- Eigenschaft FormatDisplay, B-23
- Eigenschaft KnobHeight, B-24
- Eigenschaft KnobPicture, B-24
- Eigenschaft KnobWidth, B-25
- Eigenschaft LargeChange, B-25
- Eigenschaft Locked, B-26
- Eigenschaft Offset, B-27
- Eigenschaft Picture, B-28
- Eigenschaft Precision, B-30
- Eigenschaft PushButton, B-32
- Eigenschaft RawMax, B-32
- Eigenschaft RawMin, B-32
- Eigenschaft ScaleMode, B-34
- Eigenschaft ShowErrorBoxes, B-36
- Eigenschaft ShowMinMax, B-36
- Eigenschaft SmallChange, B-37
- Eigenschaft StretchMode, B-38
- Eigenschaft Style, B-39
- Eigenschaft TagSource, B-39
- Eigenschaft Text, B-40
- Eigenschaft Ticks, B-40
- Eigenschaft TrueCaption, B-40, B-42
- Eigenschaft TrueColor, B-41
- Eigenschaft Value, B-43
- Eigenschaft WriteMode, B-43
- Eigenschaft ZeroPad, B-46

Eigenschaften

- Activated, B-1
- Alignment, B-2
- Appearance, B-3
- AutoConnect, B-3
- AutoConnectTimeout, B-4
- BackColor, B-5
- bDiagBuffOK, B-5
- bEngineConnected, B-6
- BorderStyle, B-6
- Caption, B-7
- ControlEngine, B-10
- DataType, B-11
- DefaultDeadband, B-12

Eigenschaften

- DefaultUpdateRate, B-13
- Direction, B-14
- DisplayFormatButtons, B-15
- DisplayHelpButton, B-15
- DisplayHelpOnEventButton, B-16
- DisplayLowerPanel, B-17
- DisplayUpdateButton, B-17
- DisplayUpperPanel, B-18
- DisplayValue, B-18
- Enabled, B-19
- EnableSort, B-19
- Factor, B-20
- FalseCaption, B-21
- FalseColor, B-21
- FalsePicture, B-22
- FomatDisplay, B-23
- Font, B-22
- ForeColor, B-23
- KnobHeight, B-24
- KnobPicture, B-24
- KnobWidth, B-25
- LargeChange, B-25
- Locked, B-26
- Max und Min, B-26
- Methode AboutBox, B-1
- Methode Connect, B-7
- Methode ConnectName, B-8
- Methode ConnectObject, B-9
- Methode Disconnect, B-14
- Methode PopUpHelp, B-29
- Methode PopUpHelpOnEvent, B-29
- Methode PropertyChangedName, B-30
- Methode PropertyChangedObject, B-31
- Methode ReadMultiVariables, B-33
- Methode ReadVariable, B-33
- Methode SelectEvent, B-35
- Methode Update, B-42
- Methode WriteMultiVariables, B-44
- Methode WriteVariable, B-45
- MultipleEngines, B-27
- Offset, B-27
- PCName, B-28
- Picture, B-28
- Precision, B-30
- PushButton, B-32
- RawMax, B-32
- RawMin, B-32
- ScaleMode, B-34
- ShowErrorBoxes, B-36
- ShowMinMax, B-36

Eigenschaften

- SIMATIC Controls
 - Button, 6-2 – 6-6
 - Data Control, 5-29 – 5-30
 - DBuffer Control, 7-7 – 7-8
- SmallChange, B-37
- StretchMode, B-38
- Style, B-39
- TagSource, B-39
- Text, B-40
- Ticks, B-40
- TrueCaption, B-40, B-42
- TrueColor, B-41
- Value, B-43
- WriteMode, B-43
- WriteNow Methode, B-44
- ZeroPad, B-46
- Eigenschaften Caption, B-7
- Eigenschaften Max und Min, B-26
- Eigenschaften und Methoden
 - Button Control, 6-9
 - Edit Control, 6-18
 - Label Control, 6-26
 - Slider Control, 6-34
- Eingänge, Speicherbereiche PE und E von
 - S7-Automatisierungssystemen, A-2
- Eingänge und Ausgänge, S7-Automatisierungssysteme, A-2
- Eingänge von S7-Automatisierungssystemen
 - Auswirkungen des Zyklus auf Eingänge und Ausgänge, A-2
 - Zugreifen auf den Speicherbereich PE, A-2
- Einstellen der Sprache, D-5
- Englisch, Ändern, D-5
- Ereignis Change, C-1
- Ereignis Click, C-1
- Ereignis ConnectionError, C-1
 - Fehlercodes (Data Control), 5-30
- Ereignis DblClick, C-2
- Ereignis Error, C-2
- Ereignis KeyDown, C-3
- Ereignis KeyPress, C-5
- Ereignis KeyUp, C-5
- Ereignis MouseDown, C-6
- Ereignis MouseMove, C-7
- Ereignis MouseUp, C-8
- Ereignis ValueChanged, C-9
- Ereignisse
 - Beispielprogramm, 5-19 – 5-22
 - Button Control, 6-10
 - Change, C-1

Ereignisse

- Click, C-1
- ConnectionError, C-1
- Data Control, 5-30
- DbClick, C-2
- DBuffer Control, 2-3, 7-1
- Edit Control, 6-19
- Ergänzen zum Data Control, 5-15
- Error, C-2
- KeyDown, C-3
- KeyPress, C-5
- KeyUp, C-5
- Label Control, 6-26
- MouseDown, C-6
- MouseMove, C-7
- MouseUp, C-8
- Slider Control, 6-35
- ValueChanged, C-9
- Ereignistabelle, Beispielprogramm, 5-18
- Ergänzen einer Verbindung,
zum Data Control, 5-12

F

- Fehlerbehebung
 - DCOM, E-20
 - keine gültige Autorisierung, 3-3
- Fehlercodes
 - Data Control, 5-30
 - Edit Control, 6-20
- Französisch, Ändern, D-5
- Fremd-ActiveX-Control, 2-3 – 2-6
 - OPC Controls, 2-6 – 2-8
- Fremd-Container, F-2 – F-4
- Fremd-Control, 8-5 – 8-7

G

- Gemeinsames Nutzen von Daten in mehreren Applikationen
 - OPC Controls, 2-3 – 2-5, 2-6 – 2-8
 - SIMATIC Controls, 2-3, 5-1 – 5-12

H

- Handelsübliche Anwendungen,
 - OPC Controls, 2-6

I

Installation

- Autorisierung, 3-3 – 3-5
- Deinstallieren der Autorisierung, 3-4
- Installieren der Autorisierung für die Software SIMATIC Computing, 3-4
 - Richtlinien, 3-4
- Installieren und Deinstallieren, 3-5 – 3-7
- Kopierschutz, 3-3 – 3-5
 - Deinstallieren der Autorisierung, 3-4
 - Übertragen der Autorisierung, 3-4
- Systemanforderungen, 2-3
- Übertragen der Autorisierung, 3-4
- Installieren der Autorisierung für die Software SIMATIC Computing
 - Deinstallieren der Autorisierung, 3-4
 - Übertragen der Autorisierung, 3-4
 - Vorgehensweise, 3-4
 - Richtlinien, 3-4
- Integration von dezentralen Anwendungen (DCOM), Client und Server, E-1 – E-20
- Integrieren von dezentralen Anwendungen (DCOM), lokal und entfernt, 4-1 – 4-7

K

- Kommunikation, lokal und entfernt (DCOM), Client und Server, E-1 – E-20
- Kommunikation im Netz,
 - lokal und entfernt, 4-1 – 4-7
 - Client und Server, E-1 – E-20
- Kommunizieren,
 - lokal und entfernt (DCOM), 4-1 – 4-7
- Konfigurationswerkzeug, Einstellen der lokalen Control Engine, D-6
- Kopierschutz, 3-3 – 3-5
 - Deinstallieren der Autorisierung, 3-4
 - Übertragen der Autorisierung, 3-4

L

Label Control

- Beschreibung, 6-21
- Eigenschaften
 - Alignment, B-2
 - Appearance, B-3
 - BackColor, B-5

Label Control

- BorderStyle, B-6
- Caption, B-7
- Enabled, B-19
- Font, B-22
- ForeColor, B-23
- StretchMode, B-38
- Style, B-39

- Eigenschaften und Methoden, 6-26

Ereignisse, 6-26

- Change, C-1
- Click, C-1
- DbClick, C-2
- Error, C-2
- MouseDown, C-6
- MouseMove, C-7
- MouseUp, C-8

- Methoden, AboutBox, B-1

- Schaltfläche in der Funktionsleiste, 6-21

Leistungsfähigkeit

- Data Control für kritische Daten, F-4
- Verbinden/Verbindung trennen, B-4, B-8, B-9, B-14, F-5

- Liesmich-Datei, Richtlinien für die Autorisierung von WinLC, 3-3

- Löschen einer Verbindung, 5-12

M

- Megahertz (MHz), Systemanforderungen, 2-3

- Merker, S7-Speicherbereich (M), A-2

- Methode AboutBox, B-1

- Methode Connect, B-7

- Methode ConnectName, B-8

- Methode ConnectObject, B-9

- Methode Disconnect, B-14

- Methode Methode WriteVariable, B-45

- Methode PCName, B-28

- Methode PopUpHelp, B-29

- Methode PopUpHelpOnEvent, B-29

- Methode PropertyChangedName, B-30

- Methode PropertyChangedObject, B-31

- Methode ReadMultiVariables, B-33

- Methode ReadVariable, B-33

- Methode SelectEvent, B-35

- Methode Update, B-42

- Methode WriteMultiVariables, B-44

- Methode WriteNow, B-44

Methoden

- Beispiele, 5-24 – 5-30

- ConnectObject, B-9

- Data Control, 5-29

Methoden

- DBuffer Control, 7-7

- Disconnect, B-14

- Methode WriteMultiVariables, B-44

- Methode WriteVariable, B-45

- PopUpHelp, B-29

- PopUpHelpOnEvent, B-29

- PropertyChangedName, B-30

- PropertyChangedObject, B-31

- ReadMultiVariables, B-33

- ReadVariable, B-33

- S7DiagBF Control, SelectEvent, B-35

- WriteNow, B-44

- MHz, Systemanforderungen, 2-3

- Modus Design, 8-7

- MultipleEngines, B-27

N

- Name des OPC-Servers, 2-7

- NOT-AUS, 1-1, 2-1, 8-6

O**OLE**

- siehe auch* SIMATIC Computing

- OLE for Process Control. *siehe* OPC

- OPC Controls, 2-6 – 2-8

- OPC-Dokumentation, 2-6, 2-7

- SIMATIC Controls, 5-1 – 5-12

- Button, 6-4

- Edit, 6-11 – 6-22

- Label, 6-21

- Slider, 6-27 – 6-29

- SoftContainer, 8-1

- OPC, 2-6 – 2-8

- Client-Applikation, 2-3, 2-7

- Data Control, 2-3

- Gemeinsames Nutzen von Daten in mehreren Applikationen, 2-3 – 2-5, 2-7

- Gruppe, Schnittstellen, 2-7

- Name des Servers, 2-7

- OPC-Dokumentation, 2-6

- Schnittstellen der Gruppe, 2-7

- Schnittstellen des Servers, 2-7

- Server, 2-3

- Name, 2-7

- Schnittstellen, 2-7

- Verwendung mit SIMATIC Computing, 2-7

- Optionen, Ändern der Sprache, D-5

P

Pentium, Systemanforderungen, 2-3
Peripheriebereiche der Ein- und Ausgänge von S7-Automatisierungssystemen, Adressierung, A-2
Personal Computer (PC), Systemanforderungen, 2-3
Produktübersicht, 2-3 – 2-5
 OPC (Ole for Process Control), 2-6 – 2-8
ProgID, 2-7
Programmable Logic Controller (PLC). *siehe* Control Engine
Programmatic Identifier, 2-7
Programmieren, Verbinden/Verbindung trennen, B-4, B-8, B-9, B-14, F-5
Programmierung
 anwenderspezifische ActiveX Controls, F-7 – F-10
 Beispielprogramme
 Bedienpanel, 1-4 – 1-10
 Microsoft Excel, 1-15 – 1-19
 SoftContainer, 1-19 – 1-25
 STEP 7 Programm, 1-3
 weitere Controls (VBScrollbar), 1-12 – 1-15
 Data Control für kritische Daten, F-4
 Richtlinien für Container, F-2
 S7-Datentypen in VB und C, A-5
 Timer, F-4
Prozeßdaten
 OPC, 2-6 – 2-8
 SIMATIC Controls, 5-1 – 5-12
 Button, 6-4 – 6-8
 Data, 5-1 – 5-13
 Edit, 6-11 – 6-22
 Label, 6-21 – 6-25
 Slider, 6-27 – 6-29
 Zugreifen, 2-3 – 2-5
Prozessor (CPU), PC-Anforderungen, 2-3

R

RAM, Systemanforderungen, 2-3
Richtlinien
 Anwenderspezifische ActiveX Controls, F-7 – F-10
 Auswirkungen des Zyklus auf Eingänge und Ausgänge, A-2
 Autorisierung für WinLC, 3-3
 Beispielprogramme
 Bedienpanel, 1-4 – 1-10
 Microsoft Excel, 1-15 – 1-19

SoftContainer, 1-19 – 1-25

STEP 7 Programm, 1-3

weitere Controls (VBScrollbar), 1-12 – 1-15

Containers, F-2

Data Control für kritische Daten, F-4

NOT-AUS, 1-1, 2-1, 8-6

Verbinden/Verbindung trennen, B-4, B-8, B-9, B-14, F-5

Verwenden von Timer in Visual Basic, A-6, F-4

Zugreifen auf Daten vom Datentyp STRING und CHAR (BSTR), A-9

Zugreifen auf die Speicherbereiche PE und PA, A-2

S

S7-Automatisierungssysteme

 OPC Controls, 2-6 – 2-8

 SIMATIC Controls

 Button, 6-4 – 6-8

 Edit, 6-11 – 6-22

 Label, 6-21 – 6-25

 Slider, 6-27 – 6-29

 Speicherbereiche, A-2

 Zyklus, A-2

S7-Datentypen

 ARRAY, A-6 – A-8

 DATE, A-7

 in Visual Basic und C, A-5

 S5TIME, A-8

 STRING, A-9

 TIME, A-10

 TIME_OF_DAY, A-10

S7DiagBF Control

 Eigenschaften

 bDiagBuffOK, B-5

 bEngineConnected, B-6

 ControlEngine, B-10

 DisplayFormatButtons, B-15

 DisplayHelpButton, B-15

 DisplayHelpOnEventButton, B-16

 DisplayLowerPanel, B-17

 DisplayUpdateButton, B-17

 DisplayUpperPanel, B-18

 EnableSort, B-19

 FormatDisplay, B-23

 Methode PopUpHelp, B-29

 Methode PopUpHelpOnEvent, B-29

 Methode SelectEvent, B-35

 Methode Update, B-42

 Ereignisse, Click, C-1

- Server (OPC), 2-3, 2-6 – 2-7
 - Schnittstellen, 2-7
 - Server-Name, 2-7
- Server-Konfiguration (DCOM), E-4 – E-13
- Setup-Programm
 - Autorisierung, 3-3 – 3-4
 - Speicheranforderungen, 2-3
- SIMATIC Computing
 - Anforderungen an das Betriebssystem, 2-3
 - Anforderungen an den Computer, 2-3
 - Button Control
 - Beschreibung, 6-4
 - Eigenschaften, 6-2 – 6-6
 - Schaltfläche in der Funktionsleiste, 6-4
 - Data
 - Beschreibung, 5-1
 - Schaltfläche in der Funktionsleiste, 5-1
 - Data Control
 - Eigenschaften, 5-29 – 5-30
 - Ereignisse, 5-30
 - Fehlercodes, 5-30
 - Deinstallieren der Autorisierung, 3-4
 - Edit Control
 - Beschreibung, 6-11
 - Schaltfläche in der Funktionsleiste, 6-11
 - Eigenschaften
 - Button, 6-2 – 6-6
 - Data, 5-29 – 5-30
 - Ereignisse, Data, 5-30
 - Fehlercodes, Data, 5-30
 - Installation
 - Deinstallieren der Autorisierung, 3-4
 - Kopierschutz, 3-3 – 3-5
 - Systemanforderungen, 2-3
 - Übertragen der Autorisierung, 3-4
 - Vorgehensweise, 3-5 – 3-7
 - Installation, Autorisierung, 3-3 – 3-5
 - Konfigurationswerkzeug, D-6
 - Label Control
 - Beschreibung, 6-21
 - Schaltfläche in der Funktionsleiste, 6-21
 - OPC Controls, 2-7 – 2-9
 - Server, 2-7
 - Produktübersicht, 2-1 – 2-8
 - S7-Speicherbereiche, A-2
 - SIMATIC Controls
 - Beschreibung, 2-4 – 2-6
 - Button, 6-4 – 6-8
 - Data Control, 5-1 – 5-12
 - Edit, 6-11 – 6-22
 - Label, 6-21 – 6-25
 - Slider, 6-27 – 6-29
- SIMATIC Controls
 - Slider Control, Beschreibung, 6-27
 - SoftContainer, 8-1
 - Speicheranforderungen, 2-3
 - Systemanforderungen, 2-3
 - Button, 6-4 – 6-8
 - Button Control
 - Schaltfläche in der Funktionsleiste, 6-4
 - Data
 - Beschreibung, 5-1
 - Schaltfläche in der Funktionsleiste, 5-1
 - Data Control, 5-1
 - DBuffer, Beschreibung, 2-3, 7-1
 - DBuffer Control, 2-3, 7-1
 - Edit, 6-11 – 6-22
 - Edit Control
 - Beschreibung, 6-11
 - Schaltfläche in der Funktionsleiste, 6-11
 - Eigenschaften, Activated, B-1
 - gemeinsames Nutzen von Daten in mehreren Applikationen, 2-3
 - Label, 6-21 – 6-25
 - Label Control
 - Beschreibung, 6-21
 - Schaltfläche in der Funktionsleiste, 6-21
 - Slider, 6-27 – 6-29
 - Slider Control, Beschreibung, 6-27
 - verwendet in SIMATIC Computing, 2-3
- Slider Control, 2-4
 - Beschreibung, 2-4, 6-27
 - Eigenschaften
 - BackColor, B-5
 - Direction, B-14
 - DisplayValue, B-18
 - Enabled, B-19
 - Factor, B-20
 - ForeColor, B-23
 - KnobHeight, B-24
 - KnobPicture, B-24
 - KnobWidth, B-25
 - LargeChange, B-25
 - Locked, B-26
 - Max und Min, B-26
 - Offset, B-27
 - Picture, B-28
 - RawMax, B-32
 - RawMin, B-32
 - ScaleMode, B-34
 - ShowMinMax, B-36
 - SmallChange, B-37
 - StretchMode, B-38
 - Style, B-39

- Eigenschaften und Methoden, 6-34
 - Ticks, B-40
 - Value, B-43
 - Ereignisse, 6-35
 - Change, C-1
 - Click, C-1
 - DbClick, C-2
 - Error, C-2
 - KeyDown, C-3
 - KeyPress, C-5
 - KeyUp, C-5
 - MouseDown, C-6
 - MouseMove, C-7
 - MouseUp, C-8
 - Methoden, AboutBox, B-1
 - Slot PLC. *siehe* Control Engine
 - SoftContainer, 8-1 – 8-7
 - Anlegen eines Prozeß-Formulars, 8-2 – 8-4
 - Beispielprogramm, 1-19 – 1-25
 - Betriebszustand, 8-6 – 8-8
 - Funktionsleisten, 8-2 – 8-4
 - Prozeß-Formular, 8-4 – 8-6
 - Symbole, 8-2 – 8-4
 - Übersicht, 8-2 – 8-4
 - Software PLC. *siehe* Control Engine
 - Software-Installation
 - Autorisierung für SIMATIC Computing, 3-3 – 3-5
 - Installieren und Deinstallieren, 3-5 – 3-7
 - Deinstallieren der Autorisierung für SIMATIC Computing, 3-3 – 3-5
 - Übertragen der Autorisierung für SIMATIC Computing, 3-3 – 3-5
 - Speicheranforderungen, 2-3
 - Speicherbereiche PE und PA, A-2
 - Speicherbereiche von S7-Automatisierungssystemen
 - Adressierung, A-2
 - OPC Controls, 2-6 – 2-8
 - SIMATIC Controls, 5-1 – 5-12
 - Button, 6-4 – 6-8
 - Edit, 6-11 – 6-22
 - Label, 6-21 – 6-25
 - Slider, 6-27 – 6-29
 - Spracheinstellung, für WinAC, D-5
 - Symboldatei
 - Anlegen, 9-6
 - Control Engine ohne Symbole, 9-8
 - Data Control, 5-5
 - Einfügen eines Programms oder einer Control Engine, 9-7
 - Symboldatei
 - Einrichten für lokalen oder entfernten Zugriff, 9-10
 - entfernte Control Engine, 4-5
 - Komponenten, 9-5
 - mehrere Control Engines, 4-6, 9-3
 - Symboldatei-Konfigurator, 9-1 – 9-7
 - Arbeiten mit Symbolen, 9-5
 - Symboldateien, lokale und entfernte Control Engines, 4-1 – 4-7
 - Systemanforderungen, 2-3
- T**
- Tabellenkalkulationen, gemeinsames Nutzen von Daten über OPC, 2-6
 - Technische Daten
 - OLE for Process Control, 2-6, 2-7
 - Systemanforderungen, 2-3
- U**
- Übersicht
 - OPC Controls, 2-3, 2-6 – 2-8
 - SIMATIC Computing, 2-3 – 2-5
 - SIMATIC Controls, 2-3 – 2-5
 - Übertragen der Autorisierung für SIMATIC Computing, 3-3 – 3-5
 - Übertragen der Autorisierung für WinLC, Richtlinien, 3-3
 - Überwachen und Ändern von Daten
 - OPC Controls, 2-6 – 2-8
 - SIMATIC Controls, 5-1 – 5-12
 - Button, 6-4 – 6-8
 - Edit, 6-11 – 6-22
 - Ereignisse (Data), 5-30
 - Fehlercodes (Data), 5-30
 - Label, 6-21 – 6-25
 - Slider, 6-27 – 6-29
 - Speicherbereiche von S7-Automatisierungssystemen, A-2
- V**
- Verbinden mit Daten über Data Control, 2-4 – 2-6
 - Verbinden von Daten über OPC, 2-6 – 2-8
 - Verbindungstabelle
 - Beispielprogramm, 5-17
 - Data Control, 5-16

Visual Basic
 anwenderspezifische ActiveX Controls,
 F-6 – F-10
 Beispielprogramm, F-7 – F-10
 Beispielprogramm, Bedienpanel, 1-4 – 1-10
 Beispielprogramme
 Microsoft Excel, 1-15 – 1-19
 SoftContainer, 1-19 – 1-25
 STEP 7 Programm, 1-3
 weitere Controls (VBScrollbar),
 1-12 – 1-15
 BSTR für Daten vom Datentyp STRING
 und CHAR, A-9
 Data Control für kritische Daten, F-4
 Datentypen, A-5
 Richtlinien für Container, F-2
 Timer (Richtlinien), A-6, F-4
 Verbinden/Verbindung trennen, B-4, B-8,
 B-9, B-14, F-5
 Vorgehensweisen
 Autorisierung für die Software SIMATIC
 Computing, Hinzufügen einer Autorisie-
 rung, 3-4
 Deinstallieren der Autorisierung, 3-4
 Deinstallieren der Software, 3-6
 Installieren der Autorisierung für die Soft-
 ware SIMATIC Computing, 3-4
 Richtlinien, 3-4
 Installieren der Autorisierung für die Soft-
 ware WinLC, Deinstallieren einer Autori-
 sierung, 3-4
 Installieren der Software SIMATIC Compu-
 ting, 3-5
 Zugreifen auf OPC Server, 2-7

W

Warnung, NOT-AUS, 1-1, 2-1, 8-6
 Warnungen, Timer in Visual Basic, A-6, F-4
 Wechseln der Betriebszustände des
 SoftContainer, 8-7
 WinACDefault, D-6

Windows Logic Controller (WinLC)
 siehe Control Engine
 WinLC
 Extras, Sprache, D-5
 Übertragen der Autorisierung, 3-4
 WriteNow, B-44

Z

Zähler, S7-Speicherbereich, A-2
 Zeichenfolgen der Control Engine, PG/PC-
 Schnittstelle, 5-4
 Zeiten, S7-Speicherbereich, A-2
 Zugreifen auf Daten
 eigenes Data Control, F-4
 Verbinden/Verbindung trennen, B-4, B-8,
 B-9, B-14, F-5
 Zugreifen auf den OPC-Server, 2-7
 Zugreifen auf Prozeßdaten, 2-3 – 2-5
 ActiveX Controls
 Button, 6-4 – 6-8
 Edit, 6-11 – 6-22
 Label, 6-21 – 6-25
 Slider, 6-27 – 6-29
 OPC Controls, 2-6 – 2-9
 SIMATIC Controls, 5-1 – 5-12
 Speicherbereiche von S7-Automatisie-
 rungssystemen, A-2
 Zugreifen auf Speicherbereiche, SIMATIC
 Controls, 5-1 – 5-12
 Zugreifen auf Speicherbereiche (S7)
 ActiveX Controls
 Button, 6-4 – 6-8
 Edit, 6-11 – 6-22
 Label, 6-21 – 6-25
 Slider, 6-27 – 6-29
 OPC Controls, 2-6 – 2-9
 Speicherbereiche von S7-Automatisie-
 rungssystemen, A-2
 Zuordnen einer Variablen in Visual Basic, 1-6
 Zyklus von S7-Automatisierungssystemen, A-2

An
Siemens AG
A&D AS E 81
Östliche Rheinbrückenstr. 50
76181 Karlsruhe

Absender:

Ihr Name: _____
Ihre Funktion: _____
Ihre Firma: _____
Straße: _____
Ort: _____
Telefon: _____

Bitte kreuzen Sie Ihren zutreffenden Industriezweig an:

- | | |
|--|--|
| ☐ Automobilindustrie | ☐ Pharmazeutische Industrie |
| ☐ Chemische Industrie | ☐ Kunststoffverarbeitung |
| ☐ Elektroindustrie | ☐ Papierindustrie |
| ☐ Nahrungsmittel | ☐ Textilindustrie |
| ☐ Leittechnik | ☐ Transportwesen |
| ☐ Maschinenbau | ☐ Andere _____ |
| ☐ Petrochemie | |



Vorschläge und Anmerkungen zur Anwenderdokumentation

Ihre Anmerkungen und Vorschläge helfen uns, die Qualität und Benutzbarkeit unserer Dokumentation zu verbessern. Bitte füllen Sie diesen Fragebogen bei der nächsten Gelegenheit aus und senden Sie ihn an Siemens zurück.

Geben Sie bitte bei den folgenden Fragen Ihre persönliche Bewertung mit Werten von 1 = gut bis 5 = schlecht an.

- | | | |
|----|---|--------------------------|
| 1. | Entspricht der Inhalt Ihren Anforderungen? | ☐ |
| 2. | Sind die benötigten Informationen leicht zu finden? | ☐ |
| 3. | Sind die Texte leicht verständlich? | ☐ |
| 4. | Entspricht der Grad der technischen Einzelheiten Ihren Anforderungen? | ☐ |
| 5. | Wie bewerten Sie die Qualität der Abbildungen und Tabellen? | ☐ |

Falls Sie auf konkrete Probleme gestoßen sind, erläutern Sie diese bitte in den folgenden Zeilen:
